Scalable, Flexible and Active Learning on Distributions

Dougal J. Sutherland

CMU-CS-16-128
September 2016

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Jeff Schneider, Chair
Maria-Florina Balcan
Barnabás Póczos
Arthur Gretton, University College London

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

This research was sponsored by the National Science Foundation under grant number IIS0911032; the Chicago Department of Energy under grant number DESC0002607; Science Applications International Corp under grants P010127116T0003, P010132891T0005, and P01032981T0006; Sandia National Laboratories under grant number PO1486778; and the US Air Force Research Laboratory under grants FA875010C0210, FA87501220324, and FA87501420244.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: kernel methods, approximate embeddings, statistical machine learning, nonparametric statistics, two-sample testing, active learning.

To my parents.

Abstract

A wide range of machine learning problems, including astronomical inference about galaxy clusters, natural image scene classification, parametric statistical inference, and detection of potentially harmful sources of radiation, can be well-modeled as learning a function on (samples from) distributions. This thesis explores problems in learning such functions via kernel methods, and applies the framework to yield state-of-the-art results in several novel settings.

One major challenge with this approach is one of computational efficiency when learning from large numbers of distributions: the computation of typical methods scales between quadratically and cubically, and so they are not amenable to large datasets. As a solution, we investigate approximate embeddings into Euclidean spaces such that inner products in the embedding space approximate kernel values between the source distributions. We provide a greater understanding of the standard existing tool for doing so on Euclidean inputs, random Fourier features. We also present a new embedding for a class of information-theoretic distribution distances, and evaluate it and existing embeddings on several real-world applications.

The next challenge is that the choice of distance is important for getting good practical performance, but how to choose a good distance for a given problem is not obvious. We study this problem in the setting of two-sample testing, where we attempt to distinguish two distributions via the maximum mean divergence, and provide a new technique for kernel choice in these settings, including the use of kernels defined by deep learning-type models.

In a related problem setting, common to physical observations, autonomous sensing, and electoral polling, we have the following challenge: when observing samples is expensive, but we can choose where we would like to do so, how do we pick where to observe? We give a method for a closely related problem where we search for instances of patterns by making point observations.

Throughout, we combine theoretical results with extensive empirical evaluations to increase our understanding of the methods.

Acknowledgements

To start with, I want to thank my advisor, Jeff Schneider, for so ably guiding me through this long process. When I began my PhD, I didn't have a particular research agenda or project in mind. After talking to Jeff a few times, we settled on my joining an existing project: applying this crazy idea of distribution kernels to computer vision problems. Obviously, the hunch that this project would turn into something I'd be interested in working on more worked out. Throughout that project and the ones I've worked on, Jeff has been instrumental in asking the right questions to help me realize when I've gone off in a bad direction, in suggesting better alternatives, and in thinking pragmatically about problems, using the best tools for the job and finding the right balance between empirical and theoretical results.

Barnabás Póczos also basically would have been an advisor had he not still been a postdoc when I started my degree. His style is a great complement to Jeff's, caring about many of the same things but coming at them from a somewhat different direction. Nina Balcan knew the right connections from the theoretical community, which I haven't fully explored enough yet. I began working in more depth with Arthur Gretton over the past six months or so, and it's been both very productive and great fun, which is great news since I'm now very excited to move on to a postdoc with him.

Several of my labmates have also been key to everything I've done here. Liang Xiong helped me get up and running when I started my degree, spending hours in my office showing me how things worked and how to make sense of the results we got. Yifei Ma's enthusiasm about widely varying research ideas was inspiring. Junier Oliva always knew the right way to think about something when I got stuck. Tzu-Kuo Huang spent an entire summer thinking about distribution learning with me and eating many, many plates of chicken over rice. Roman Garnett is a master of Gaussian processes and appreciated my disappointment in Pittsburgh pizza. I never formally collaborated much with Ben, María, Matt, Samy, Sibi, or Xuezhi, but they were always fun to talk about ideas with. The rest of the Auton Lab, especially Artur Dubrawski, made brainstorming meetings something to look forward to whenever they happened.

Outside the lab, Michelle Ntampaka was a joy to collaborate with on applications to cosmology problems, even when she was too embarrassed to show me her code for the experiments. The rest of the regular Astro/Stat/ML group also helped fulfill, or at least feel like I was fulfilling, my high school dreams of learning about the universe. Fish Tung made crazy things work. The XDATA crew made perhaps-too-frequent drives to DC and long summer days packed in a small back room worth it, especially frequent Phronesis collaborators Ben Johnson, who always had an interesting problem to think about, and Casey King, who always knew an interesting person to talk to.

Karen Widmaier, Deb Cavlovich, and Catherine Copetas made everything run smoothly: without them, not only would nothing ever actually happen, but the things that did happen would be far less pleasant. Jarod Wang and Predrag Punosevac kept the lab machines going, despite my best efforts to crash, overload, or otherwise destroy them.

Other, non-research friends also made this whole endeavor worthwhile. Alejandro Carbonara always had jelly beans, Ameya Velingker made multiple spontaneous trips across state lines, Aram Ebtekar single-handedly and possibly permanently destroyed my sleep schedule, Dave Kurokawa was a good friend (there's no time to explain why), Shayak Sen received beratement

with grace, and Sid Jain gave me lots of practice for maybe having a teenager of my own one day. Listing friends in general a futile effort, but here's a few more of the Pittsburgh people without whom my life would have been worse: Alex, Ashique, Brendan, Charlie, Danny, the Davids, Goran, Jay-Yoon, Jenny, Jesse, John, Jon, Junxing, Karim, Kelvin, Laxman, Nic, Nico, Preeti, Richard, Ryan, Sarah, Shriphani, Vagelis, and Zack, as well as the regular groups for various board/tabletop games, everyone else who actually participated in departmental social events, and machine learning conference buddies. Of course, friendship is less constrained by geography than it once was: Alex Burka, James Smith, and Tom Eisenberg were subjected to frequent all-day conversations, the Board of Shadowy Figures and the PTV Mafia group provided much pleasant distraction, and I spent more time on video chats with Jamie McClintock, Luke Collin, Matt McLaughlin, and Tom McClintock than was probably reasonable.

My parents got me here and helped keep me here, and even if my dad thinks I didn't want him at my defense, I fully acknowledge that all of it is only because of them. My brother Ian and an array of relatives (grandparents, aunts, uncles, cousins, and the more exotic ones like first cousins once removed and great-uncles) are also a regular source of joy, whether I see them several times a year or a handful of times a decade. Thank you.

Contents

Chapter 1

Introduction

Traditional machine learning approaches focus on learning problems defined on vectors, mapping whatever kind of object we wish to model to a fixed number of real-valued attributes. Though this approach has been very successful in a variety of application areas, choosing natural and effective representations can be quite difficult.

In many settings, we wish to perform machine learning tasks on objects that can be viewed as a collection of lower-level objects or more directly as samples from a distribution. For example:

- Images can be thought of as a collection of local patches (Section 5.3); similarly, videos are collections of frames.
- The total mass of a galaxy cluster can be predicted based on the positions and velocities of individual galaxies (Section 5.1).
- The photons recieved by a small radiation sensor can be used to classify the presence of harmful radioactive material (Section 5.4).
- Support for a political candidate among various demographic groups can be estimated by learning a regression model from electoral districts of individual voters to district-level support for political candidates (Flaxman, Y.-X. Wang, et al. 2015).
- Documents are made of sentences, which are themselves composed of words, which themselves can be seen as being represented by sets of the contexts in which they appear (Section 8.3).
- Parametric statistical inference problems learn a function from sample sets to model parameters (Section 5.2).
- Expectation propagation techniques relay on maps from sample sets to messages normally computed via expensive numerical integration (Jitkrittum, Gretton, et al. 2015).
- Causal arrows between distributions can be estimated from samples (Lopez-Paz et al. 2015).

In order to use traditional techniques on these collective objects, we must create a single vector that represents the entire set. Though there are various ways to summarize a set as a vector, we can often discard less information and require less effort in feature engineering by operating directly on sets of feature vectors.

One method for machine learning on sets is to consider them as samples from some unknown underlying probability distribution over feature vectors. Each example then has its own distribution: if we are classifying images as sets of patches, each image is defined as a distribution over patch features, and each class of clusters is a set of patch-level feature distributions. We can then define a kernel based on statistical estimates of a distance between probability distributions. Letting $\mathcal{X} \subseteq \mathbb{R}^d$ denote the set of possible feature vectors, we thus define a kernel $k : 2^{\mathcal{X}} \times 2^{\mathcal{X}} \to \mathbb{R}$. This lets us perform classification, regression, anomaly detection, clustering, low-dimensional embedding, and any of many other applications with the well-developed suite of kernel methods. Chapter 2 discusses various such kernels and their estimators; Chapter 5 gives empirical results on several problems.

When used for a learning problem with N training items, however, typical kernel methods require operating on an $N \times N$ kernel matrix, which requires far too much computation to scale to datasets with a large number of instances. One way to avoid this problem is through approximate embeddings $z : \mathcal{X} \to \mathbb{R}^D$, à la Rahimi and Recht (2007), such that $z(x)^\mathsf{T} z(y) \approx k(x, y)$. Chapter 3 gives some new results in the theory of random Fourier embeddings, while Chapter 4 uses them as a tool in developing embeddings for several distributional kernels, which are also evaluated empirically in Chapter 5.

Chapter 6 moves to the related problem of two-sample testing. Here, we are given two sample sets X and Y, and we wish to test the hypothesis that X and Y were generated from the same distribution. This problem, closely related to classification, has many practical applications; one primary method for doing so is based on the maximum mean discrepancy (MMD) between the distributions. This method relies on a base kernel; Chapter 6 develops and evaluates a new method for selecting these kernels, including complex kernels based on deep learning.

Chapter 7 addresses the application of this type of complex functional classifier to an active search problem. Consider finding polluted areas in a body of water, based on point measurements. We wish to, given an observation budget, adaptively choose where we should make these observations in order to maximize the number of regions we can be confident are polluted. If our notion of "pollution" is defined simply by a threshold on the mean value of a univariate measurement, Y. Ma, Garnett, et al. (2014) give a natural selection algorithm based on Gaussian process inference. If, instead, our sensors measure the concentrations of several chemicals, the vector flow of water current, or other such more complicated data, we can instead apply a classifier to a region and consider the problem of finding regions that the classifier marks as relevant.

1.1 Summary of contributions

- Chapter 2 mostly establishes the framework with which we will discuss learning on distributions. Section 2.4.2 includes a mildly novel analysis not yet published.[1]
- Chapter 3 improves the theoretical understanding of the random Fourier features of Rahimi and Recht (2007). (Based on Sutherland and Schneider 2015.)
- Section 4.3 gives an approximate embedding for a new class of distributional distances. (Based on Sutherland, J. B. Oliva, et al. 2016.)
- Chapter 5 provides empirical studies for the application of distributional distances to practical problems. (Based on Póczos, Xiong, Sutherland, et al. 2012; Sutherland, Xiong,

[1] This was developed with Tzu-Kuo (TK) Huang.

et al. 2012; Ntampaka, Trac, Sutherland, Battaglia, et al. 2015; Jin 2016; Jin et al. 2016; Sutherland, J. B. Oliva, et al. 2016; Ntampaka, Trac, Sutherland, Fromenteau, et al. in press.)

- Chapter 6 develops and evaluates a new method for kernel selection in two-sample testing based on the MMD distributional distance. (Work not yet published.[2])
- Chapter 7 presents and analyzes a method for the novel problem setting of *active pointillistic pattern search*, using point observations to observe regional patterns. (Based on Y. Ma, Sutherland, et al. 2015.)
- The `skl-groups` package, overviewed in Appendix A, provides efficient implementations of several of the methods for learning on distributions discussed in this thesis.

[2] Done in collaboration with Fish Tung, Aaditya Ramdas, Heiko Strathmann, Alex Smola, and Arthur Gretton.

Chapter 2

Learning on distributions

As discussed in Chapter 1, we consider the problem of learning on probability distributions. Specifically: let $\mathcal{X} \subseteq \mathbb{R}^d$ be the set of observable feature vectors, $\mathcal{S}$ the set of possible sample sets (all finite subsets of $\mathcal{X}$), and $\mathcal{P}$ the set of probability distributions under consideration. We then perform machine learning on samples from distributions by:

1. Choosing a distance on distributions $\rho : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$.
2. Defining a Mercer kernel $k : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ based on ρ.
3. Estimating k based on the observed samples as $\hat{k} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$, which should itself be a kernel on $\mathcal{S}$.
4. Using $\hat{k}$ in a standard kernel method, such as an SVM or a Gaussian Process, to perform classification, regression, collective anomaly detection, or other machine learning tasks.

Certainly, this is not the only approach to learning on distributions. Some distributional learning methods do not directly compare sample sets to one another, but rather compare their elements to a class-level distribution (Boiman et al. 2008). Given a distance ρ, one can naturally use k-nearest neighbor models (Póczos, Xiong, and Schneider 2011; Kusner et al. 2015), or Nadaraya-Watson–type local regression models (J. B. Oliva, Póczos, et al. 2013; Póczos, Rinaldo, et al. 2013) with respect to that distance. In this thesis, however, we focus on kernel methods as a well-studied, flexible, and empirically effective approach to a broad variety of learning problems.

We typically assume that every distribution in $\mathcal{P}$ has a density with respect to the Lebesgue measure, and slightly abuse notation by using distributions P, Q and their densities p, q interchangeably.

2.1 Distances on distributions

We will define kernels on distributions by first defining distances ρ between them.

2.1.1 Distance frameworks

We first present four general frameworks for distances on distributions. These are each broad categories of distances containing (or related to) several of the concrete distance families we employ.

L_r metrics One natural way to compute distances between distributions is the L_r metric between their densities, for order $r \geq 1$:

$$L_r(p, q) \coloneqq \left(\int_{\mathcal{X}} |p(x) - q(x)|^r \, \mathrm{d}x \right)^{1/r} .$$

Note that the limit $r = \infty$ yields the distance $L_\infty(p, q) = \sup_{x \in \mathcal{X}} |p(x) - q(x)|$.

f-divergences For any convex function f with $f(1) = 0$, the f-divergence of P to Q is

$$D_f(P \| Q) \coloneqq \int_{\mathcal{X}} f\left(\frac{p(x)}{q(x)} \right) q(x) \, \mathrm{d}x.$$

This class is sometimes called "Csiszár f-divergences", after Csiszár (1963). Sometimes the requirement of convexity or that $f(1) = 0$ is dropped. Note that these functions are not in general symmetric or respecting of the triangle inequality. They do, however, satisfy $D_f(P \| P) = 0$, when f is strictly convex at 1 $D_f(P \| Q) \geq 0$, and are jointly convex:

$$D_f(\lambda P + (1 - \lambda) P' \| \lambda Q + (1 - \lambda) Q') \leq \lambda D_f(P \| Q) + (1 - \lambda) D_f(P' \| Q').$$

In fact, the only metric f-divergences are multiples of the total variation distance, discussed shortly (Khosravifard et al. 2007) — though e.g. the Hellinger distance is the square of a metric. For an overview, see e.g. Liese and Vajda (2006).

α-β divergences The following somewhat less-standard divergence family, defined e.g. by Póczos, Xiong, Sutherland, et al. (2012) generalizing the α-divergence of Amari (1985), is also useful. Given two real parameters α, β, $D_{\alpha,\beta}$ is defined as

$$D_{\alpha,\beta}(P \| Q) \coloneqq \int p^\alpha(x) \, q^\beta(x) \, p(x) \, \mathrm{d}x.$$

$D_{\alpha,\beta}(P \| Q) \geq 0$ for any α, β; $D_{\alpha,-\alpha}(P \| P) = 1$. Note also that $D_{\alpha,-\alpha}$ has the form of an f-divergence with $t \mapsto t^{\alpha+1}$, though this does not satisfy $f(1) = 0$ and is convex only if $\alpha \notin (-1, 0)$.

Integral probability metrics Many useful metrics can be expressed as *integral probability metrics* (IPMs, Müller 1997):

$$\rho_{\mathfrak{F}}(P, Q) \coloneqq \sup_{f \in \mathfrak{F}} \left| \int f \, \mathrm{d}P - \int f \, \mathrm{d}Q \right|,$$

where $\mathfrak{F}$ is some family of functions $f : \mathcal{X} \to \mathbb{R}$. Note that $\rho_{\mathfrak{F}}$ satisfies $\rho_{\mathfrak{F}}(P, P) = 0$, $\rho_{\mathfrak{F}}(P, Q) = \rho_{\mathfrak{F}}(Q, P)$, and $\rho_{\mathfrak{F}}(P, Q) \leq \rho_{\mathfrak{F}}(P, R) + \rho_{\mathfrak{F}}(R, Q)$ for any $\mathfrak{F}$, and is thus always a pseudometric; the remaining metric property of distinguishability, $(\rho_{\mathfrak{F}}(P, Q) = 0) \implies (P = Q)$, depends on $\mathfrak{F}$. Sriperumbudur et al. (2009) give an overview.

2.1.2 Specific distributional distances

The various distributional distances below can often be represented in one or more of these frameworks. Many more such distances exist; here we mainly discuss the ones used in this thesis, along with a few others of interest. Figure 2.1 gives a visual illustration of several of the distances considered here.

L_2 distance The L_2 distance is one of the most common metrics used on distributions. It can also be represented as $D_{1,0} - 2D_{0,1} + D_{-1,2}$.

Total variation distance The total variation distance (TV) is such an important distance that it is sometimes referred to simply as "the statistical distance." It can be defined as

$$\mathrm{TV}(P, Q) = \sup_A |P(A) - Q(A)|,$$

where A ranges over every event in the underlying σ-algebra. It can also be represented as $\frac{1}{2} L_1(P, Q)$, as an f-divergence with $t \mapsto |t-1|$, and as an IPM with (among other classes) $\mathfrak{F} = \{f : \sup_{x \in \mathcal{X}} f(x) - \inf_{x \in \mathcal{X}} f(x) \le 1\}$ (Müller 1997). Note that TV is a metric, and $0 \le \mathrm{TV}(P, Q) \le 1$.

The total variation distance is closely related to the "intersection distance", most commonly used on histograms (Cha and Srihari 2002):

$$\int_{\mathcal{X}} \min(p(x), q(x)) \,\mathrm{d}x = \int_{\mathcal{X}} \tfrac{1}{2} \left(p(x) + q(x) - |p(x) - q(x)|\right) \mathrm{d}x = 1 - \mathrm{TV}(P, Q).$$

Kullback-Leibler divergence The Kullback-Leibler (KL) divergence is defined as

$$\mathrm{KL}(P \| Q) := \int_{\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \,\mathrm{d}x.$$

For discrete distributions, the KL divergence bears a natural information theoretic interpretation as the expected excess code length required to send a message for P via the optimal code for Q. It is nonnegative, and zero iff $P = Q$ almost everywhere; however, $\mathrm{KL}(P\|Q) \ne \mathrm{KL}(Q\|P)$ in general. Note also that if there is any point with $p(x) > 0$ and $q(x) = 0$, $\mathrm{KL}(P\|Q) = \infty$.

Applications often use a symmetrization by averaging with the dual:

$$\mathrm{SKL}(P, Q) := \tfrac{1}{2} \left(\mathrm{KL}(P\|Q) + \mathrm{KL}(Q\|P)\right).$$

This is also sometimes called Jeffrey's divergence, though that name is also sometimes used to refer to the Jensen-Shannon divergence (below), so we avoid it. SKL does not satisfy the triangle inequality.

KL can be viewed as a f divergence, with one direction corresponding to $t \mapsto t \log t$ and the other to $t \mapsto -\log t$; SKL is thus an f divergence with $t \mapsto \frac{1}{2}(t-1)\log t$.

Jensen-Shannon divergence The Jensen-Shannon divergence is based on $\textsc{kl}$:

$$\textsc{js}(P, Q) := \tfrac{1}{2}\,\textsc{kl}\left(P \middle\| \frac{P+Q}{2}\right) + \tfrac{1}{2}\,\textsc{kl}\left(Q \middle\| \frac{P+Q}{2}\right),$$

where $\frac{P+Q}{2}$ denotes an equal mixture between P and Q. $\textsc{js}$ is clearly symmetric, and in fact $\sqrt{\textsc{js}}$ satisfies the triangle inequality. Note also that $0 \le \textsc{js}(P, Q) \le \log 2$. It gets its name from the fact that it can be written as the Jensen difference of the Shannon entropy:

$$\textsc{js}(P, Q) = \mathrm{H}\left[\frac{P+Q}{2}\right] - \frac{\mathrm{H}[P] + \mathrm{H}[Q]}{2},$$

a view which allows a natural generalization to more than two distributions. Non-equal mixtures are also natural, but of course asymmetric. For more details, see e.g. Martins et al. (2009).

Rényi-α divergence The Rényi-α divergence (Rényi 1961) generalizes $\textsc{kl}$ as

$$\textsc{r}_\alpha(P\|Q) := \frac{1}{\alpha-1} \log \int p(x)^\alpha q(x)^{1-\alpha}\, \mathrm{d}x;$$

note that $\lim_{\alpha\to1} \textsc{r}_\alpha(P\|Q) = \textsc{kl}(P\|Q)$, though $\alpha = 1$ is not defined. $\textsc{r}_\alpha$ is typically used for $\alpha \in (0, 1) \cup (1, \infty)$; for $\alpha < 0$, it can be negative. Like $\textsc{kl}$, $\textsc{r}_\alpha$ is asymmetric; we similarly define a symmetrization

$$\textsc{sr}_\alpha(P, Q) := \tfrac{1}{2}\left(\textsc{r}_\alpha(P\|Q) + \textsc{r}_\alpha(Q\|P)\right).$$

$\textsc{sr}_\alpha$ does not satisfy the triangle inequality.

$\textsc{r}_\alpha$ can be represented based on an α-β divergence: $\textsc{r}(P\|Q) = \frac{1}{\alpha-1} \log D_{\alpha-1,1-\alpha}(P\|Q)$.

A Jensen-Rényi divergence, defined by replacing $\textsc{kl}$ with $\textsc{r}_\alpha$ in the definition of $\textsc{js}$, has also been studied (Martins et al. 2009), but we will not consider it here.

Tsallis-α divergence The Tsallis-α divergence, named after Tsallis (1988) but previously studied by Havrda and Charvát (1967) and Daróczy (1970), provides a different generalization of $\textsc{kl}$:

$$\textsc{t}_\alpha(P\|Q) := \frac{1}{\alpha-1}\left(\int p(x)^\alpha q(x)^{1-\alpha}\, \mathrm{d}x - 1\right).$$

Again, $\lim_{\alpha\to1} \textsc{t}_\alpha(P\|Q) = \textsc{kl}(P\|Q)$, and of course $\textsc{t}_\alpha = \frac{1}{\alpha-1}\left(D_{\alpha-1,1-\alpha}(P\|Q) - 1\right)$. Because of its close relation to $\textsc{r}_\alpha$, we will not use it further.

Hellinger distance The square of the Hellinger distance $\textsc{h}$ is defined as

$$\textsc{h}^2(P, Q) := \tfrac{1}{2} \int \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 \mathrm{d}x = 1 - \int \sqrt{p(x)\, q(x)}\, \mathrm{d}x.$$

$\textsc{h}^2$ can be expressed as an f-divergence with either $t \mapsto \frac{1}{2}(\sqrt{t} - 1)^2$ or $t \mapsto 1 - \sqrt{t}$; it is also closely related to an α-β divergence as $\textsc{h}^2(P, Q) = 1 - D_{-1/2,1/2}$. $\textsc{h}$ is a metric, and is bounded in $[0, 1]$. It is proportional to the L_2 difference between $\sqrt{p}$ and $\sqrt{q}$, which yields the bounds $\textsc{h}^2(P, Q) \le \textsc{tv}(P, Q) \le \sqrt{2}\, \textsc{h}(P, Q)$.

χ^2 divergence There are many distinct definitions of the χ^2 divergence. We do not directly use any in this thesis, but the most common versions are:

$$\begin{aligned}
\chi^2_P(P\|Q) &:= \int \left(\frac{p(x)}{q(x)} - 1\right)^2 q(x)\,\mathrm{d}x = \int \frac{(p(x)-q(x))^2}{q(x)}\,\mathrm{d}x = \int \frac{p(x)^2}{q(x)}\,\mathrm{d}x - 1 \\
\chi^2_N(P\|Q) &:= \int \frac{(p(x)-q(x))^2}{p(x)}\,\mathrm{d}x = \int \frac{q(x)^2}{p(x)}\,\mathrm{d}x - 1 \\
\chi^2_S(P,Q) &:= \frac{1}{2}\int \frac{(p(x)-q(x))^2}{p(x)+q(x)}\,\mathrm{d}x \\
\chi^2_A(P,Q) &:= 2\left(1 - \int \frac{p(x)\,q(x)}{p(x)+q(x)}\,\mathrm{d}x\right).
\end{aligned}$$

$\chi^2_P(P\|Q)$ is an f-divergence using either $t \mapsto (t-1)^2$ or $t \mapsto t^2 - 1$, used e.g. by Liese and Vajda (2006); it is sometimes called the Pearson divergence or similar, and is often used in hypothesis testing of multinomial data. $\chi^2_N(P\|Q)$, termed the Neyman divergence e.g. by Cressie and Read (1984), is its dual: $\chi^2_N(P\|Q) = \chi^2_P(Q\|P)$. Neither is commonly used in learning on distributions.

$\chi^2_S(P,Q)$ is a symmetric variant of these distances; its use on discrete distributions, especially histograms, is common in computer vision (Puzicha et al. 1997; Zhang et al. 2006).

Vedaldi and Zisserman (2012) use the kernel $k_{\chi^2}(P,Q) := 2\int \frac{p(x)\,q(x)}{p(x)+q(x)}\,\mathrm{d}x$, sometimes called the additive χ^2 kernel (e.g. by Grisel et al. 2016), which corresponds to the distance χ^2_A. Despite a claim to the contrary by Vedaldi and Zisserman (2012), it is not equal to χ^2_S.

Earth mover's distance The earth mover's distance ($\textsc{emd}_\rho$) is defined for a metric ρ as

$$\textsc{emd}_\rho(P,Q) := \inf_{R \in \Gamma(P,Q)} \mathbb{E}_{(X,Y)\sim R}\left[\rho(X,Y)\right], \tag{2.1}$$

where $\Gamma(P,Q)$ is the set of joint distributions with marginals P and Q. It is also called the first *Wasserstein distance*, or the *Mallows distance*. When $(\mathcal{X},\rho)$ is separable (in the topological sense), it is also equal to the *Kantorovich metric*, which is the IPM with $\mathfrak{F} = \{f : \|f\|_L \le 1\}$, where $\|f\|_L := \sup\{|f(x)-f(y)|/\rho(x,y) \mid x \ne y \in \mathcal{X}\}$ is the Lipschitz semi-norm. Edwards (2011) gives some historical details and proves the equality in a more general setting.

For discrete distributions, EMD can be computed via linear programming, and is popular in the computer vision community (e.g. Rubner et al. 2000; Zhang et al. 2006).

Cuturi (2013) proposes a distance called the Sinkhorn distance, which replaces $\Gamma(P,Q)$ in (2.1) with a constraint that the KL divergence of the distribution from the independent be less than some parameter α. This both allows for much faster computation of the distance on discrete distributions and, in certain problems, yields learning models that outperform those based on the full EMD.

Maximum mean discrepancy The maximum mean discrepancy, called the MMD (Sriperumbudur, Gretton, et al. 2010; Gretton, Borgwardt, et al. 2012) is defined by embedding distributions into a reproducing kernel Hilbert space (RKHS; for a detailed overview see Berlinet and Thomas-Agnan 2004). Let κ be the kernel associated with some RKHS $\mathcal{H}$ with feature map $\varphi : \mathcal{X} \to \mathcal{H}$, also denoted $\varphi(x) = \kappa(x, \cdot)$, such that $\langle \varphi(x), \varphi(y) \rangle_{\mathcal{H}} = \kappa(x, y)$. We can then map a distribution P to its mean embedding $\mu_{\mathcal{H}}(P) := \mathbb{E}_{X \sim P}[\varphi(X)]$, and define the distance between distributions as the distance between their mean embeddings:

$$\textsc{mmd}_\kappa(P, Q) := \|\mu_{\mathcal{H}}(P) - \mu_{\mathcal{H}}(Q)\|_{\mathcal{H}}.$$

$\textsc{mmd}_\kappa$ can also be viewed as an IPM with $\mathfrak{F} = \{f \in \mathcal{H} \mid \|f\|_{\mathcal{H}} \le 1\}$, where $\|f\|_{\mathcal{H}}$ is the norm in $\mathcal{H}$. (If $f \in \mathcal{H}$, $f(\cdot) = \sum_{i=1}^{\infty} \alpha_i \kappa(x_i, \cdot)$ for some points $x_i \in \mathcal{X}$ and weights $\alpha_i \in \mathbb{R}$; $\|f\|_{\mathcal{H}}^2 = \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j)$.) In fact, the function f achieving the supremum is known as the *witness function*, and is achieved by $f = \mu_P - \mu_Q$.

The mean embedding always exists when the base kernel κ is bounded, in which case $\textsc{mmd}_\kappa$ is a pseudometric; full metricity requires a *characteristic* κ. See Sriperumbudur, Gretton, et al. (2010) and Gretton, Borgwardt, et al. (2012) for details.

Szabó et al. (2015) proved learning-theoretic bounds on the use of ridge regression with MMD.

2.2 Estimators of distributional distances

We now discuss methods for estimating different distributional distances ρ.

The most obvious estimator of most distributional distances is perhaps the plug-in approach: first perform density estimation, and then compute distances between the density estimates. These approaches suffer from the problem that the density is in some sense a nuisance parameter for the problem of distance estimation, and density estimation is quite difficult, particularly in higher dimensions.

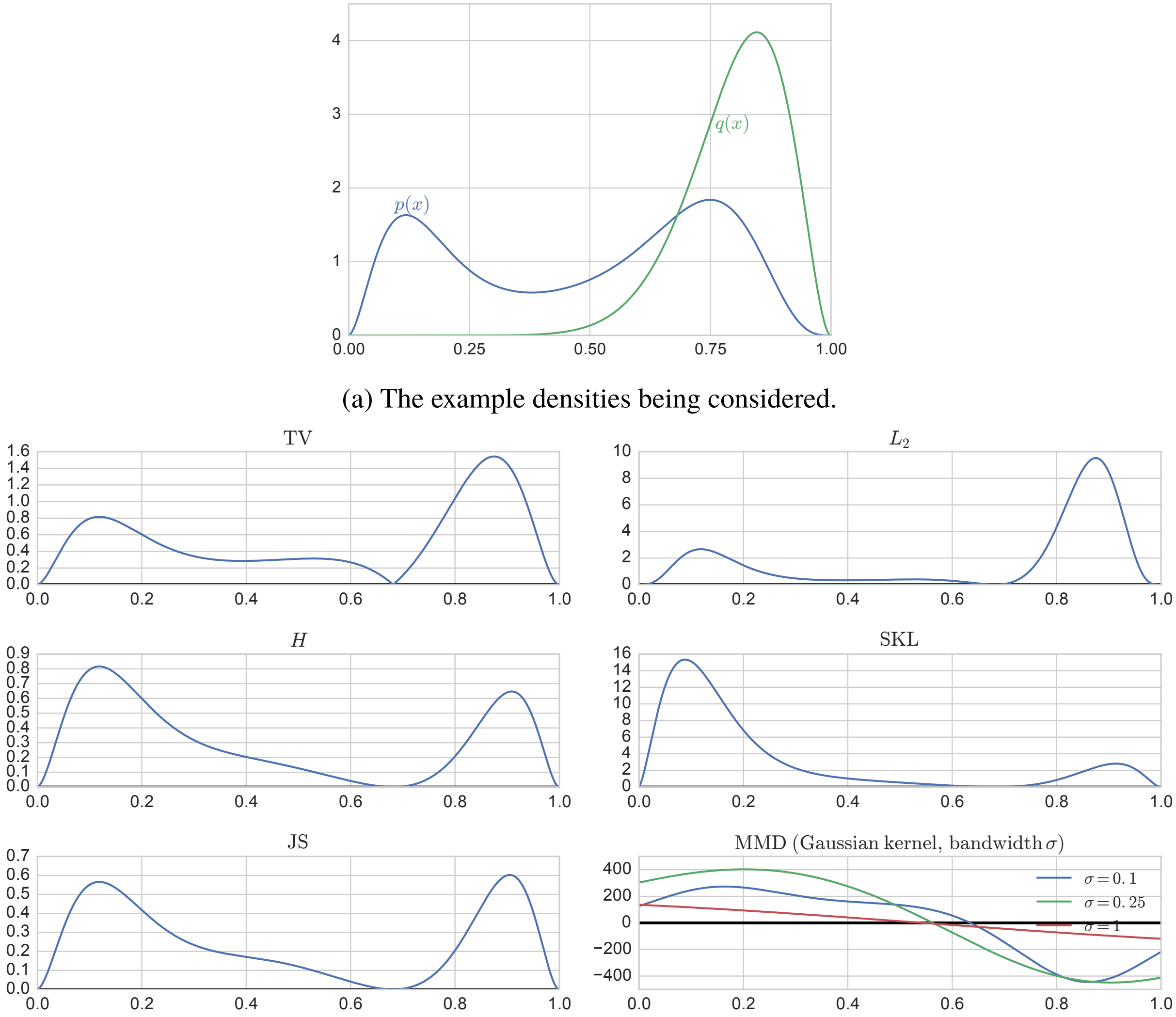
Some of the methods below are plug-in methods; others correct a plug-in estimate, or use inconsistent density estimates in such a way that the overall divergence estimate is consistent.

Parametric models Closed forms of some distances are available for certain distributions:

- For members of the same exponential family, closed forms of the Bhattacharyya kernel (corresponding to Hellinger distance) and certain other kernels of the form $D_{\alpha-1,\alpha}$ were computed by Jebara et al. (2004). Nielsen and Nock (2011) give closed forms for all $D_{\alpha-1,1-\alpha}$, allowing the computation of $\textsc{r}_\alpha$, $\textsc{t}_\alpha$, and related divergences, as well as the KL divergence via $\lim_{\alpha \to 1} D_{\alpha-1,1-\alpha}$.
- For Gaussian distributions, Muandet, Schölkopf, et al. (2012) compute the closed form of MMD for a few base kernels. Sutherland (2015) also conjectures a form for the Euclidean EMD and gives bounds.
- For mixture distributions, L_2 and MMD can be computed based on the inner products between the components by simple linearity arguments. For mixtures specifically of Gaussians, F. Wang et al. (2009) obtain the quadratic ($\textsc{r}_2$) entropy, which allows the computation of Jensen-Rényi divergences for $\alpha = 2$.

For cases when a closed form does not exist, numerical integration may be necessary, often obviating the computational advantages of this approach.

It is thus possible to fit a parametric model to each distribution and compute distances between the fits; this is done for machine learning applications e.g. by Jebara et al. (2004) and Moreno



(a) The example densities being considered.

(b) The functions being integrated for some of the distances. For example, the TV image shows $\frac{1}{2}|p(x) - q(x)|$.

Figure 2.1: An illustration of some of the distributional distances considered here.

et al. (2004). In practice, however, we rarely know that a given parametric family is appropriate, and so the use of parametric models introduces unavoidable approximation error and bias.

Histograms One common method for representing distributions is the use of histograms; many distances ρ are then simple to compute, typically in $O(m)$ time for m-bin histograms. The prominent exception to that is EMD, which requires $O(m^3 \log m)$ time for exact computation (e.g. Rubner et al. 2000), though in some settings $O(m)$ approximations are available (Shirdhonkar and Jacobs 2008) and as previously mentioned, the related Sinkhorn distance can be computed quite quickly (Cuturi 2013). MMD also requires approximately $O(m^2)$ computation for typical histograms.

The main disadvantages of histograms are their poor performance in even moderate dimensions, and the fact that (for most ρs) choosing the right bin size is both quite important and quite difficult, since nearby bins do not affect one another. Histogram density estimators also give non-optimal rates for density estimation (Wasserman 2006), and provide technical difficulties in establishing consistent estimation as bin sizes decrease (Gretton and Györfi 2010).

Vector quantization An improvement over standard histograms, popular in computer vision, is to instead quantize distributions to group points by their nearest *codeword* from a dictionary, often learned via k-means or a similar algorithm. This method is known as the *bag of words* (BOW) approach and was popularized by Leung and Malik (2001). This method empirically scales to much higher dimensions than the histogram approach, but suffers from similar problems related to the hard assignment of sample points to bins.

Grauman and Darrell (2007) use multiple resolutions of histograms to compute distances, helping somewhat with the issue of choosing bin sizes.

Kernel density estimation Perhaps the most popular form of general-purpose nonparametric density estimation is kernel density estimation (KDE). KDE results in a mixture distribution, which allow $O(n^2)$ exact computation of plug-in MMD and L_2 for certain density kernels. Selection of the proper bandwidth, however, is a significant issue.

Singh and Póczos (2014) show exponential concentration for a particular plug-in estimator for a broad class of functionals including L_p, $D_{\alpha,\beta}$, and f-divergences as well as JS, though they do not discuss computational issues of the estimator, which in general requires numerical integration.

Krishnamurthy et al. (2014) correct a plug-in estimator for L_2 and R_α divergences by estimating higher order terms in the von Mises expansion; one of their estimators is computationally attractive and optimal for smooth distributions, while another is optimal for a broader range of distributions but requires numerical integration.

***k*-NN density estimator** The k-NN density estimator provides the basis for another family of estimators. These estimators require k-nearest neighbor distances within and between the sample sets. Much research has been put into data structures for efficient approximate nearest neighbor computation (e.g. Beygelzimer et al. 2006; Muja and Lowe 2009; Andoni and Razenshteyn 2015; Naidan et al. 2015), though in high dimensions the problem is quite difficult and brute-force pairwise computation may be the most efficient method. Plug-in methods require k to grow with sample size for consistency, typically at a rate around $\sqrt{n}$, which makes computation more difficult.

Q. Wang et al. (2009) give a simple, consistent k-NN KL divergence estimator. Póczos and Schneider (2011) give a similar estimator for $D_{\alpha-1,1-\alpha}$ and show consistency; Póczos, Xiong, Sutherland, et al. (2012) generalize to $D_{\alpha,\beta}$. This family of estimators is consistent with a fixed k, though convergence rates are not known.

Moon and Hero (2014a,b) propose an f-divergence estimator based on ensembles of plug-in estimators, and show the distribution is asymptotically Gaussian. (Their estimator requires neither convex f nor $f(1) = 0$.)

Mean map estimators A natural estimator of $\langle \mu_{\mathcal{H}}(P), \mu_{\mathcal{H}}(Q) \rangle_{\mathcal{H}}$ is simply the mean of the pairwise kernel evaluations between the two sets, $\frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \kappa(X_i, Y_j)$; this is the inner product between embeddings of the empirical distributions of the two samples. The estimator $\frac{1}{n} \sum_{i=1}^{n} \kappa(X_i, Y_i)$ allows use in the streaming setting. We can then estimate MMD via $\|x - y\|^2 = \langle x, x \rangle + \langle y, y \rangle - 2\langle x, y \rangle$ (Gretton, Borgwardt, et al. 2012). Section 6.1 gives much more detailed on variations of these estimators of MMD.

Muandet, Fukumizu, et al. (2014) proposed biasing the estimator of MMD to obtain smaller variance via the idea of Stein shrinkage (1956). Ramdas and Wehbe (2015) showed the efficacy of this approach for independence testing.

Other approaches Nguyen et al. (2010) provide an estimator for f-divergences (requiring convex f but not $f(1) = 0$) by solving a convex program. When an RKHS structure is imposed, it requires solving a general convex program with dimensionality equal to the number of samples, so the estimator is quite computationally expensive.

Sriperumbudur et al. (2012) estimate the L_1-EMD via a linear program.

K. Yang et al. (2014) estimate f- and R_α divergences by adaptively partitioning both distributions simultaneously. Their Bayesian approach requires MCMC and is computationally expensive, though it does provide a posterior over the divergence value which can be useful in some settings.

2.3 Kernels on distributions

We consider two methods for defining kernels based on distributional distances ρ. Proposition 1 of Haasdonk and Bahlmann (2004) shows that both methods always create positive definite kernels iff ρ is isometric to an L_2 norm, i.e. there exist a Hilbert space $\mathcal{H}$ and a mapping $\Phi : \mathcal{X} \to \mathcal{H}$ such that $\rho(P, Q) = \|\Phi(P) - \Phi(Q)\|$. Such metrics are also called *Hilbertian.*[1]

For distances that do not satisfy this property, we will instead construct an indefinite kernel as below and then "correct" it, as discussed in Section 2.4.1.

[1] Note that if ρ is Hilbertian, Proposition 1 (ii) of Haasdonk and Bahlmann (2004) shows that $-\rho^{2\beta}$ is conditionally positive definite for any $0 \le \beta \le 1$; by a classic result of Schoenberg (1938), this implies that ρ^β is also Hilbertian. We will use this fact later.

The first method is to create a "linear kernel" k such that $\rho^2(P, Q) = k(P, P) + k(Q, Q) - 2k(P, Q)$, so that the RKHS with inner product k has metric ρ. Note that, while distances are translation-invariant, inner products are not; we must thus first choose some origin B. Then

$$k_{\text{lin}}^{(B)}(P, Q) := \tfrac{1}{2}\left(\rho^2(P, B) + \rho^2(Q, B) - \rho^2(P, Q)\right) \tag{2.2}$$

is a valid kernel for any B iff ρ is Hilbertian. If ρ is defined for the zero measure, it is often most natural to use that as the origin; in the cases it is used below, it is easy to verify that k_{lin}^{0} is a valid kernel inducing the relevant distance despite issues of whether $\rho(P, 0)$ is defined.

We can also use ρ in a *generalized* RBF *kernel*: for a bandwidth parameter $\sigma > 0$,

$$k_{\textsc{rbf}}^{(\sigma)}(x, y) := \exp\left(-\frac{1}{2\sigma^2}\rho^2(p, q)\right). \tag{2.3}$$

The L_2 distance is clearly Hilbertian; $k_{\text{lin}}^{(0)}(P, Q) = \int p(x) q(x) \, \mathrm{d}x$.

Fuglede (2005) shows that $\sqrt{\textsc{tv}}$, $\textsc{h}$, and $\sqrt{\textsc{js}}$ are Hilbertian.[2]

- For $\sqrt{\textsc{tv}}$, $k_{\text{lin}}^{(0)}(P, Q) = \frac{1}{2}\left(1 - \textsc{tv}(P, Q)\right)$ since $\textsc{tv}(P, 0) = \frac{1}{2}\lVert P \rVert_1 = \frac{1}{2}$.
- For $\textsc{h}$, $k_{\text{lin}}^{(0)}(P, Q) = 1 - \frac{1}{2}\textsc{h}^2(P, Q) = \frac{1}{2} + \int \sqrt{p(x)\, q(x)} \mathrm{d}x$, but the halved Bhattacharyya affinity $k(P, Q) = \frac{1}{2}\int \sqrt{p(x)\, q(x)} \mathrm{d}x$ is more natural.
- For $\sqrt{\textsc{js}}$, $k_{\text{lin}}^{(O)}(P, Q) = \frac{1}{2}\left(\mathrm{H}\left[\frac{P+O}{2}\right] + \mathrm{H}\left[\frac{Q+O}{2}\right] - \mathrm{H}\left[\frac{P+Q}{2}\right] - \mathrm{H}[O]\right)$.

Topsøe (2000) shows that χ_S is Hilbertian; $k_{\text{lin}}^{(0)}(P, Q) = 1 - \frac{1}{2}\chi_S^2(P, Q)$. The computer vision community sometimes uses as a kernel simply $-\chi_S^2(P, Q)$, which is only conditionally positive definite Zhang et al. (2006). χ_A is also Hilbertian, as shown by Vedaldi and Zisserman (2012) using the result of Fuglede (2005).

Gardner et al. (2015) show that EMD is Hilbertian for the unusual choice of ground metric $\rho(x, y) = \mathbb{1}(x \neq y)$. EMD is probably not Hilbertian in most cases for Euclidean base distance: Naor and Schechtman (2007) prove that Euclidean EMD on distributions supported on a grid in $\mathbb{R}^2$ does not embed in L_1, which since L_2 embeds into L_1 (Bretagnolle et al. 1966) means that EMD on that grid does not embed in L_2. It is thus extremely likely that this also implies L_2-EMD on continuous distributions over $\mathbb{R}^d$ for $d \geq 2$ is not Hilbertian. The most common kernel based on EMD, however, is actually $\exp\left(-\gamma \, \textsc{emd}(P, Q)\right)$. Whether that kernel is positive definite seems to remain an open question, defined by whether $\sqrt{\textsc{emd}}$ is Hilbertian; studies that have used it in practice have not reported finding any instance of an indefinite kernel matrix (Zhang et al. 2006).

The MMD is Hilbertian by definition. The natural associated linear kernel is $k_{\text{lin}}^{(0)}(P, Q) = \langle \mu_{\mathcal{H}}(P), \mu_{\mathcal{H}}(Q) \rangle_{\mathcal{H}}$, which we term the *mean map kernel* (MMK).

2.4 Kernels on sample sets

As discussed previously, in practice we rarely directly observe a probability distribution; rather, we observe samples from those distributions. We will instead construct a kernel on sample sets,

[2] See his Theorem 2. For $\sqrt{\textsc{tv}}$, use $K_{\infty,1}$; for $\textsc{h}$, use $K_{1,\frac{1}{2}}$. For $\sqrt{\textsc{js}}$, differentiate $K_{p,1}$ around $p = 1$, following the note after the theorem.

based on an estimate of a kernel on distributions using an estimate of the base distance ρ.

We wish to estimate a kernel on N distributions $\{P^{(i)}\}_{i=1}^N$ based on an iid sample from each distribution $\{X^{(i)}\}_{i=1}^N$, where $X^{(i)} = \{X_j^{(i)}\}_{j=1}^{n_i}$, $X_j^{(i)} \in \mathbb{R}^d$. Given an estimator $\hat\rho(X^{(i)}, X^{(j)})$ of $\rho(P^{(i)}, P^{(j)})$, we estimate $k(P_i, P_j)$ with $\hat k(X^{(i)}, X^{(j)})$ by substituting $\hat\rho(X^{(i)}, X^{(j)})$ for $\rho(P^{(i)}, P^{(j)})$ in (2.2) or (2.3). We thus obtain an estimate $\hat K$ of the true kernel matrix K, where $\hat K_{i,j} = \hat k(X^{(i)}, X^{(j)})$.

2.4.1 Handling indefinite kernel matrices

Section 2.3 established that K is positive semidefinite for many distributional distances ρ, but for some, particularly SKL and SR$_\alpha$, K is indefinite. Even if K is PSD, however, depending on the form of the estimator $\hat K$ is likely to be indefinite.

In this case, for many downstream learning tasks we must modify $\hat K$ to be positive semidefinite. Chen et al. (2009) study this setting, presenting four methods to make $\hat K$ PSD:

- Spectrum clip: Set any negative eigenvalues in the spectrum of $\hat K$ to zero. This yields the nearest PSD matrix to $\hat K$ in Frobenius norm, and corresponds to the view where negative eigenvalues are simply noise.
- Spectrum flip: Replace any negative eigenvalues in the spectrum with their absolute value.
- Spectrum shift: Increase each eigenvalue in the spectrum by the magnitude of the smallest eigenvalue, by taking $\hat K + |\lambda_{\min}| I$. When $|\lambda_{\min}|$ is small, this is computationally simpler – it is easier to find $\lambda_{\min}$ than to find all negative eigenvalues, and requires modifying only the diagonal elements — but can change $\hat K$ more drastically.
- Spectrum square: Square the eigenvalues, by using $\hat K \hat K^T$. This is equivalent to using the kernel estimates as features.

We denote this operation by Π.

When test values are available at training time, i.e. in a transductive setting, it is best to perform these operations on the full kernel matrix containing both training and test points: that is, to use $\Pi\left(\begin{bmatrix} \hat K_{\text{train}} & \hat K_{\text{train,test}} \\ \hat K_{\text{test,train}} & \hat K_{\text{test}} \end{bmatrix}\right)$. (Note that $\hat K_{\text{test}}$ is not actually used by e.g. an SVM.) If the changes are performed only on the training matrix, i.e. using $\begin{bmatrix} \Pi\left(\hat K_{\text{train}}\right) & \hat K_{\text{train,test}} \\ \hat K_{\text{test,train}} & \hat K_{\text{test}} \end{bmatrix}$, which is necessary in the typical inductive setting, the resulting full kernel matrix may not be PSD, and the kernel estimates may be treated inconsistently between training and test points. This is more of an issue for a truly-indefinite kernel, e.g. one based on KL or R$_\alpha$, where the changes due to Π may be larger.

When the test values are not available, Chen et al. (2009) propose a heuristic to account for the effect of Π: for spectrum clip and flip, they find the linear transformation which maps $\hat K_{\text{train}}$ to $\Pi(\hat K_{\text{train}})$, based on the eigendecomposition of $\hat K_{\text{train}}$, and apply it to $\hat K_{\text{test,train}}$. That is, they find the P such that $\Pi(\hat K_{\text{train}}) = P \hat K_{\text{train}}$ as follows: let the eigendecomposition of $\hat K_{\text{train}}$ be $U \Lambda U^\mathsf{T}$, with eigenvalues denoted $\lambda_1, \dots, \lambda_N$. Then P is $U M U^\mathsf{T}$, with M defined as:

$$\begin{aligned} M_{\text{flip}} &:= \operatorname{diag}(\operatorname{sign}(\lambda_1), \dots, \operatorname{sign}(\lambda_N)) \qquad (2.4) \\ M_{\text{clip}} &:= \operatorname{diag}(\mathbb{1}(\lambda_1 \ge 0), \dots, \mathbb{1}(\lambda_N \ge 0)). \end{aligned}$$

For spectrum shift, no such linear transform is available, but it is easy to account for the effect of Π: simply add $|\lambda_{\min}| I$ to $\hat{K}_{\text{test}}$ as well.

In general, we find that the transductive method is better than the heuristic approach, which is better than ignoring the problem, but the size of these gaps is problem-specific: for some problems, the gap is substantial, but for others it matters little.

When performing bandwidth selection for a generalized Gaussian RBF kernel, this approach requires separately eigendecomposing each $\hat{K}_{\text{train}}$. Xiong (2013, Chapter 6) considers a different solution: rank-penalized metric multidimensional scaling according to $\hat{\rho}$, so that standard Gaussian RBF kernels may be applied to the embedded points. That work does not consider the inductive setting, though an approach similar to that of Bengio et al. (2004) is likely to be applicable.

2.4.2 Nyström approximation

When N is large, computing and operating on the full $N \times N$ kernel matrix can be quite expensive: many kernel entries must be computed and stored (or else re-computed, at significant cost per entry), and many learning techniques as well as the techniques to account for indefiniteness in the kernel estimate require $O(N^3)$ work.

One method for approaching this problem is the Nyström extension (Williams and Seeger 2000). In this method, we somehow pick $m < N$ anchor points, perhaps by uniform random sampling or by approximate leverage scores (El Alaoui and Mahoney 2015). Reordering the kernel matrix so that these m points come first, let the kernel matrix be $\begin{bmatrix} A & B \\ B^\mathsf{T} & C \end{bmatrix}$, where A is the $m \times m$ kernel matrix of the anchor points, B is the $m \times (N-m)$ matrix of kernel values from the anchor points to all other points, and C is the $(N-m) \times (N-m)$ matrix of kernel values among the other points. We fully evaluate A and B, but leave C unevaluated; our goal is to approximate it assuming that the matrix is low-rank.

Standard Nyström The Nyström method does so by assuming that K is of rank m, and using Λ as the eigenvalues for K, while approximating the $N-m$ unknown eigenvectors by $B^\mathsf{T} U \Lambda^\dagger$, where $\Lambda^\dagger$ denotes the Moore-Penrose pseudoinverse of Λ. (Here, Λ is diagonal, so the pseudoinverse coincides with the standard inverse except if any eigenvalues are zero.) Thus our approximation of K is

$$\begin{aligned}
\bar{U} &:= \begin{bmatrix} U \\ B^\mathsf{T} U \Lambda^\dagger \end{bmatrix} \\
\tilde{K} &:= \bar{U} \Lambda \bar{U}^\mathsf{T} \\
&= \begin{bmatrix} U \Lambda U^\mathsf{T} & U \Lambda \Lambda^\dagger U^\mathsf{T} B \\ B^\mathsf{T} U \Lambda^\dagger \Lambda U^\mathsf{T} & B^\mathsf{T} U \Lambda^\dagger \Lambda \Lambda^\dagger U^\mathsf{T} B \end{bmatrix} \\
&= \begin{bmatrix} A & A A^\dagger B \\ B^\mathsf{T} A^\dagger A & B^\mathsf{T} A^\dagger B \end{bmatrix}.
\end{aligned}$$

When A is nonsingular, $AA^\dagger = I$ and so the $m \times (N-m)$ portions of the kernel matrix are unchanged. Otherwise,

$$AA^\dagger = U \Lambda U^\mathsf{T} U \Lambda^\dagger U^\mathsf{T} = U \operatorname{diag}(\mathbb{1}(\lambda_1 \neq 0), \dots, \mathbb{1}(\lambda_m \neq 0)) U^\mathsf{T}$$

and so B is projected onto the image of A.

Explicit m-dimensional embeddings for the training points are then available as $\bar{U} \Lambda^{\frac{1}{2}}$, which can then be used in training models; new points can be embedded in the same m-dimensional space comparably. For a recent theoretical analysis of the effect of this approximation on kernel ridge regression, see Rudi et al. (2015).

Indefinite Nyström via SVD When A is indefinite, we need to combine the Nyström approach with some type of projection to the PSD cone, as in Section 2.4.1. Belongie et al. (2002) give an analagous method for doing so, which we present here:

Let the singular value decomposition of A be $U_\text{SVD} \Lambda_\text{SVD} V_\text{SVD}^\mathsf{T}$. Let $\bar{U}_\text{SVD} := \begin{bmatrix} U_\text{SVD} \\ B^\mathsf{T} U_\text{SVD} \Lambda_\text{SVD}^\dagger \end{bmatrix}$, and the Nyström reconstruction be

$$\begin{aligned} \tilde{K}_\text{SVD} &:= \bar{U}_\text{SVD} \Lambda_\text{SVD} \bar{U}_\text{SVD}^\mathsf{T} \\ &= \begin{bmatrix} U_\text{SVD} \Lambda_\text{SVD} U_\text{SVD}^\mathsf{T} & U_\text{SVD} \Lambda_\text{SVD} \Lambda_\text{SVD}^\dagger U_\text{SVD}^\mathsf{T} B \\ B^\mathsf{T} U_\text{SVD} \Lambda_\text{SVD}^\dagger \Lambda_\text{SVD} U^\mathsf{T} & B^\mathsf{T} U_\text{SVD} \Lambda_\text{SVD}^\dagger \Lambda_\text{SVD} \Lambda_\text{SVD}^\dagger U_\text{SVD}^\mathsf{T} B \end{bmatrix} \end{aligned}$$

To understand this approximation, define $A_\text{flip} := U \operatorname{abs}(\Lambda) U^\mathsf{T}$, where abs denotes taking the elementwise absolute value: this is the "spectrum flip" method of Chen et al. (2009). Then, we have that $U_\text{SVD} \Lambda_\text{SVD} U_\text{SVD}^\mathsf{T} = A_\text{flip}$. First, $AA^\mathsf{T} = U_\text{SVD} \Lambda_\text{SVD}^2 U_\text{SVD}^\mathsf{T}$, so (as singular values are nonnegative) its matrix square root is just $U_\text{SVD} \Lambda_\text{SVD} U_\text{SVD}^\mathsf{T}$. We also have that $AA^\mathsf{T} = U \Lambda^2 U^\mathsf{T}$, so its matrix square root can also be written $U \operatorname{abs}(\Lambda) U^\mathsf{T} = A_\text{flip}$. Because the principal square root of the PSD matrix AA^T is unique, $U_\text{SVD} \Lambda_\text{SVD} U_\text{SVD}^\mathsf{T} = A_\text{flip}$. Thus

$$\tilde{K}_\text{SVD} = \begin{bmatrix} A_\text{flip} & A_\text{flip} A_\text{flip}^\dagger B \\ B^\mathsf{T} A_\text{flip}^\dagger A_\text{flip} & B^\mathsf{T} A_\text{flip}^\dagger B \end{bmatrix}. \tag{2.5}$$

Explicit m-dimensional features are again available as $\bar{U}_\text{SVD} \Lambda_\text{SVD}^{\frac{1}{2}}$.

As long as A is nonsingular, A_flip is positive definite, and the $m \times (N-m)$ evaluations are unaffected. Again, if A is singular then B is projected onto the image of A_flip.

Consistent indefinite Nyström The last approach corresponds to, given A and B, taking the Nyström approximation with A_flip and an unmodified B. But not modifying B to account for the PSD transformation means that, if a point from the m inducing points were repeated in the $N-m$ other points, it would be treated inconsistently. We could assuage this problem with the heuristic linear transform of Chen et al. (2009) by performing the Nyström approximation based on A_flip and $P_\text{flip} B = U M_\text{flip} U^\mathsf{T} B$, where M_flip was defined in (2.4), rather than A_flip and an unmodified B.

This gives a reconstruction of

$$\begin{aligned}
\bar{U}_{\text{flip}} &:= \begin{bmatrix} U \\ (P_{\text{flip}} B)^\mathsf{T} U \Lambda_{\text{flip}}^\dagger \end{bmatrix} = \begin{bmatrix} U \\ B^\mathsf{T} U M_{\text{flip}} U^\mathsf{T} U \Lambda_{\text{flip}}^\dagger \end{bmatrix} = \begin{bmatrix} U \\ B^\mathsf{T} U \Lambda^\dagger \end{bmatrix} \\
\tilde{K}_{\text{flip}} &:= \bar{U}_{\text{flip}} \Lambda_{\text{flip}} \bar{U}_{\text{flip}}^\mathsf{T} \\
&= \begin{bmatrix} U \Lambda_{\text{flip}} U^\mathsf{T} & U \Lambda_{\text{flip}} \Lambda^\dagger U^\mathsf{T} B \\ B^\mathsf{T} U \Lambda^\dagger \Lambda_{\text{flip}} U & B^\mathsf{T} U \Lambda^\dagger \Lambda_{\text{flip}} \Lambda^\dagger U^\mathsf{T} B \end{bmatrix} \\
&= \begin{bmatrix} A_{\text{flip}} & P'_{\text{flip}} B \\ B^\mathsf{T} P'_{\text{flip}} & B^\mathsf{T} A_{\text{flip}}^\dagger B \end{bmatrix},
\end{aligned}$$

where $P'_{\text{flip}} = U \operatorname{diag}(\operatorname{sign}(\lambda_1), \dots, \operatorname{sign}(\lambda_m))\, U^\mathsf{T}$, which is the same as P_{flip} except with directions corresponding to zero eigenvalues zeroed out. Compared to the $A_{\text{flip}} A_{\text{flip}}^\mathsf{T}$ used in the equivalent place in (2.5), this flips directions corresponding to negative eigenvalues in A. The $m \times m$ known kernel values and $(N - m) \times (N - m)$ unknown kernel values are the same as in (2.5).

We can also use A_{clip} and $P_{\text{clip}} B$ to produce a similar $\tilde{K}_{\text{clip}}$.

A full experimental evaluation of these approaches to Nyström extension of indefinite kernels is an area for future work.

Chapter 3

Approximate kernel embeddings via random Fourier features

As discussed in Section 2.4.2, the kernel methods of Chapter 2 share a common drawback: solving learning problems with N distributions typically requires computing all or most of the $N \times N$ kernel matrix. Further, many of the methods of Section 2.4.1 to deal with indefinite kernels require eigendecompositions, often requiring $O(N^3)$ work. For large N, this quickly becomes impractical.

Section 2.4.2 gave one approach for countering this problem. Rahimi and Recht (2007) spurred recent interest in another method: approximate embeddings $z : \mathcal{X} \to \mathbb{R}^D$ such that $k(x, y) \approx z(x)^\mathsf{T} z(y)$. Learning primal models in $\mathbb{R}^D$ using the z features can then usually be accomplished in time linear in n, with the models on z approximating the models on k.

This chapter reviews the method of Rahimi and Recht (2007), providing some additional theoretical understanding to the original analyses. Chapter 4 will apply these techniques to the distributional setting.

3.1 Setup

Rahimi and Recht (2007) considered continuous shift-invariant kernels on $\mathbb{R}^d$, i.e. those that can be written $k(x, y) = \underline{k}(\Delta)$, where we will use $\Delta := x - y$ throughout. In this case, Bochner's theorem (1959) guarantees that the Fourier transform of $\underline{k}$ will be a nonnegative finite measure on $\mathbb{R}^d$, which can be easily normalized to a probability distribution. Thus if we define

$$\tilde{z}(x) := \sqrt{\frac{2}{D}} \left[\sin(\omega_1^\mathsf{T} x) \quad \cos(\omega_1^\mathsf{T} x) \quad \ldots \quad \sin(\omega_{D/2}^\mathsf{T} x) \quad \cos(\omega_{D/2}^\mathsf{T} x) \right]^\mathsf{T}, \; \{\omega_i\}_{i=1}^{D/2} \sim \Omega^{D/2} \tag{3.1}$$

and let $\tilde{s}(x, y) := \tilde{z}(x)^\mathsf{T} \tilde{z}(y)$, we have that

$$\tilde{s}(x, y) = \frac{2}{D} \sum_{i=1}^{D/2} \sin(\omega_i^\mathsf{T} x) \sin(\omega_i^\mathsf{T} y) + \cos(\omega_i^\mathsf{T} x) \cos(\omega_i^\mathsf{T} y) = \frac{1}{D/2} \sum_{i=1}^{D/2} \cos(\omega_i^\mathsf{T} \Delta).$$

Noting that $\mathbb{E} \cos(\omega^\mathsf{T} \Delta) = \int \Re e^{\omega^\mathsf{T} \Delta \mathrm{i}} \mathrm{d}\Omega(\omega) = \Re k(\Delta)$, where $\Re$ denotes the real part, we therefore have $\mathbb{E}\, \tilde{s}(x, y) = k(x, y)$.

$\underline{k}$ is the characteristic function of Ω, and $\underline{\tilde{s}}$ the empirical characteristic function corresponding to the samples $\{\omega_i\}$.

Rahimi and Recht (2007) also alternatively proposed

$$\breve{z}(x) := \sqrt{\frac{2}{D}} \left[\cos(\omega_1^\mathsf{T} x + b_1) \quad \dots \quad \cos(\omega_D^\mathsf{T} x + b_D)\right]^\mathsf{T} \tag{3.2}$$
$$\{\omega_i\}_{i=1}^D \sim \Omega^D, \quad \{b_i\}_{i=1}^D \overset{iid}{\sim} \mathrm{Unif}_{[0,2\pi]}^D \,.$$

Letting $\breve{s}(x, y) := \breve{z}(x)^\mathsf{T} \breve{z}(y)$, we have

$$\breve{s}(x, y) = \frac{1}{D} \sum_{i=1}^{D} \cos(\omega_i^\mathsf{T} x + b_i) \cos(\omega_i^\mathsf{T} y + b_i) = \frac{1}{D} \sum_{i=1}^{D} \cos(\omega_i^\mathsf{T}(x - y)) + \cos(\omega_i^\mathsf{T}(x + y) + 2b_i).$$

Let $t := x + y$ throughout. Since $\mathbb{E} \cos(\omega^\mathsf{T} t + 2b) = \mathbb{E}_\omega \left[\mathbb{E}_b \cos(\omega^\mathsf{T} t + 2b)\right] = 0$, we also have $\mathbb{E}\, \breve{s}(x, y) = k(x, y)$.

Thus, in expectation, both $\tilde{z}$ and $\breve{z}$ work; they are each the average of bounded, independent terms with the correct mean. For a given embedding dimension D, $\tilde{z}$ is the average of $\frac{D}{2}$ terms and $\breve{z}$ is of D, but each component of $\tilde{z}$ has lower variance; which embedding is superior is, therefore, not immediately obvious.

The academic literature seems split on the issue. In Sutherland and Schneider (2015), we examined the first 100 papers citing Rahimi and Recht (2007) in a Google Scholar search: 15 used either $\tilde{z}$ or the equivalent complex formulation, 14 used $\breve{z}$, 28 did not specify, and the remainder merely cited the paper without using the embedding. (None discussed that there was a choice between the two.) Not included in that count are Rahimi and Recht's later work (2008a,b), which used $\breve{z}$; indeed, post-publication revisions of the original paper discuss only $\breve{z}$. Practically, the three implementations of which we are aware each use $\breve{z}$: scikit-learn (Grisel et al. 2016), Shogun (Sonnenburg et al. 2010), and JSAT (Raff 2011-16).

We will show that $\tilde{z}$ is strictly superior for the popular Gaussian kernel, among others. We will also improve the uniform convergence bounds of Rahimi and Recht (2007).

3.2 Reconstruction variance

We can in fact directly find the covariance of the reconstructions:

$$\begin{aligned}
\operatorname{Cov}\left(\underline{\tilde{s}}(\Delta), \underline{\tilde{s}}(\Delta')\right) &= \frac{2}{D} \operatorname{Cov}\left(\cos(\omega^\mathsf{T}\Delta), \cos(\omega^\mathsf{T}\Delta')\right) \\
&= \frac{1}{D}\left[\mathbb{E}\left[\cos\left(\omega^\mathsf{T}(\Delta - \Delta')\right) + \cos\left(\omega^\mathsf{T}(\Delta + \Delta')\right)\right] - 2\,\mathbb{E}\left[\cos\left(\omega^\mathsf{T}\Delta\right)\right] \mathbb{E}\left[\cos\left(\omega^\mathsf{T}\Delta'\right)\right]\right] \\
&= \frac{1}{D}\left[\underline{k}(\Delta - \Delta') + \underline{k}(\Delta + \Delta') - 2\underline{k}(\Delta)\underline{k}(\Delta')\right], \qquad (3.3)
\end{aligned}$$

so that

$$\operatorname{Var} \underline{\tilde{s}}(\Delta) = \frac{1}{D}\left[1 + \underline{k}(2\Delta) - 2\underline{k}(\Delta)^2\right]. \tag{3.4}$$

Similarly,

$$\begin{aligned}
\operatorname{Cov}\left(\breve{s}(x,y), \breve{s}(x',y')\right) &= \frac{1}{D} \operatorname{Cov}\left(\cos(\omega^\mathsf{T}\Delta) + \cos(\omega^\mathsf{T} t + 2b), \cos(\omega^\mathsf{T}\Delta') + \cos(\omega^\mathsf{T} t' + 2b)\right) \\
&= \frac{1}{D} \Bigg[\operatorname{Cov}\left(\cos(\omega^\mathsf{T}\Delta), \cos(\omega^\mathsf{T}\Delta')\right) + \operatorname{Cov}\left(\cos(\omega^\mathsf{T} t + 2b), \cos(\omega^\mathsf{T} t' + 2b)\right) \\
&\qquad + \underbrace{\operatorname{Cov}\left(\cos(\omega^\mathsf{T}\Delta), \cos(\omega^\mathsf{T} t' + 2b)\right)}_{0} + \underbrace{\operatorname{Cov}\left(\cos(\omega^\mathsf{T} t + 2b), \cos(\omega^\mathsf{T}\Delta')\right)}_{0}\Bigg] \\
&= \frac{1}{D}\left[\tfrac{1}{2}\underline{k}(\Delta - \Delta') + \tfrac{1}{2}\underline{k}(\Delta + \Delta') - \underline{k}(\Delta)\underline{k}(\Delta') \right. \\
&\qquad \left. + \tfrac{1}{2}\,\mathbb{E}\cos(\omega^\mathsf{T}(t + t') + 4b) + \tfrac{1}{2}\,\mathbb{E}\cos(\omega^\mathsf{T}(t - t'))\right] \\
&= \frac{1}{D}\left[\tfrac{1}{2}\underline{k}(\Delta - \Delta') + \tfrac{1}{2}\underline{k}(\Delta + \Delta') - \underline{k}(\Delta)\underline{k}(\Delta') + \tfrac{1}{2}\underline{k}(t - t')\right], \qquad (3.5)
\end{aligned}$$

and so

$$\operatorname{Var} \breve{s}(x,y) = \frac{1}{D}\left[1 + \tfrac{1}{2}\underline{k}(2\Delta) - \underline{k}(\Delta)^2\right]. \qquad (3.6)$$

Thus $\tilde{s}$ has lower variance than $\breve{s}$ when $\underline{k}(2\Delta) < 2\underline{k}(\Delta)^2$.

Definition 3.1. *A continuous, shift-invariant positive-definite kernel function $k(x,y) = \underline{k}(\Delta)$ with $\underline{k}(0) = 1$ is* pixie *when $\underline{k}(2\Delta) \leq 2\underline{k}(\Delta)^2$ for all Δ.*

Note that the condition always holds when $\underline{k}(\Delta) \geq \frac{1}{\sqrt{2}}$, since positive-definiteness and $\underline{k}(0) = 1$ require $\underline{k}(\cdot) \leq 1$. It also trivially holds for $\underline{k}(2\Delta) \leq 0$. Once $\underline{k}$ reaches $\frac{1}{\sqrt{2}}$ in a particular direction, it then essentially must decay at least exponentially.[1]

Proposition 3.2 (Exponentiated norms)**.** *Kernels of the form $\underline{k}(\Delta) = \exp(-\gamma\|\Delta\|^\beta)$ for any norm $\|\cdot\|$ and scalars $\gamma > 0$, $\beta \geq 1$ are pixie.*

Proof. Following a simple calculation:

$$\begin{aligned}
2\underline{k}(\Delta)^2 - \underline{k}(2\Delta) &= 2\exp\left(-\gamma\|\Delta\|^\beta\right)^2 - \exp\left(-\gamma\|2\Delta\|^\beta\right) \\
&= 2\exp\left(-2\gamma\|\Delta\|^\beta\right) - \exp\left(-2^\beta\gamma\|\Delta\|^\beta\right) \\
&\geq 2\exp\left(-2\gamma\|\Delta\|^\beta\right) - \exp\left(-2\gamma\|\Delta\|^\beta\right) = \exp\left(-2\gamma\|\Delta\|^\beta\right) > 0. \qquad \square
\end{aligned}$$

For example, the Gaussian kernel uses $\|\cdot\|_2$ and $\beta = 2$, and the Laplacian kernel uses $\|\cdot\|_1$ and $\beta = 1$. The variance per dimension of embeddings for the Gaussian kernel are illustrated in Figure 3.1.

Proposition 3.3 (Matérn kernels)**.** *Define the Matérn kernel with parameters $\nu > 0$ and $\ell > 0$ as*

$$\underline{k}_{\nu,\ell}(\Delta) := \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}\|\Delta\|}{\ell}\right)^\nu K_\nu\left(\frac{\sqrt{2\nu}\|\Delta\|}{\ell}\right),$$

[1] This leads to the obscure name: such functions can have a "decreasing base" to their exponent, which might remind one of the song "Debaser" by the Pixies.

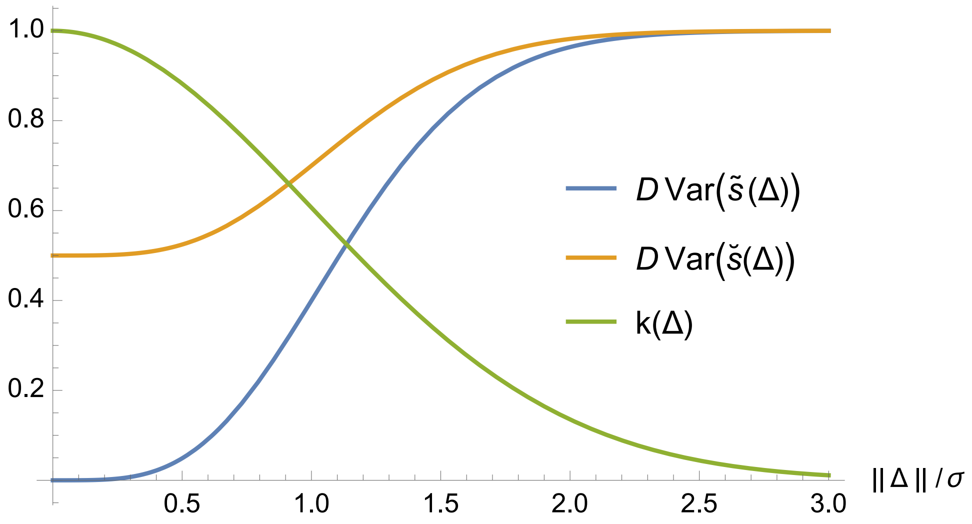


Figure 3.1: The variance per dimension of $\tilde{s}$ (blue) and $\check{s}$ (orange) for the Gaussian kernel (green).

where K_ν is a modified Bessel function of the second kind. $k_{\nu,\ell}$ is pixie for all $\nu \geq \frac{1}{2}$.

Proof. This proof is due to SKBMoore (2016). First, note that it is trivially true for $\Delta = 0$. Then define $x := \sqrt{2\nu}\|\Delta\|/\ell$. We will actually show the stricter inequality $\underline{k}(2\Delta) < \underline{k}(\Delta)^2$, which is equivalent to saying that for all $x > 0$:

$$\frac{2^{1-\nu}}{\Gamma(\nu)} (2x)^\nu K_\nu(2x) \leq \frac{2^{2-2\nu}}{\Gamma^2(\nu)} x^{2\nu} K_\nu^2(x),$$

i.e.

$$K_\nu(2x) \leq \frac{2^{1-2\nu}}{\Gamma(\nu)} x^\nu K_\nu^2(x).$$

We will need several identities about Bessel functions. These all hold for any $x > 0$:

$$K_\nu^2(x) = \frac{1}{2}\int_0^\infty \frac{1}{t} e^{-\frac{t}{2}-\frac{x^2}{t}} K_\nu\left(\frac{x^2}{t}\right) \mathrm{d}t \qquad \text{all } \nu; \text{ (DLMF, (10.32.18) with } z = \xi = x) \tag{3.7}$$

$$K_\nu(x) > 2^{\nu-1}\Gamma(\nu)e^{-x}x^{-\nu} \qquad \nu \geq \tfrac{1}{2}; \text{ (Ismail 1990, (1.4))} \tag{3.8}$$

$$K_\nu(2x) = \frac{1}{2}x^\nu \int_0^\infty \frac{1}{t^{\nu+1}} e^{-t-\frac{x^2}{t}} \mathrm{d}t \qquad \text{all } \nu; \text{ (DLMF, (10.32.10) with } z = 2x) \tag{3.9}$$

$$K_{-\nu}(x) = K_\nu(x) \qquad \text{all } \nu; \text{ (DLMF, (10.27.3))} \tag{3.10}$$

Note that Ismail (1990) shows (3.8) only for $\nu > \frac{1}{2}$, but it holds for $\nu = \frac{1}{2}$ as well by a trivial calculation, since $K_{1/2}(x) = \sqrt{\frac{\pi}{2x}}e^{-x}$ (DLMF, (10.39.2)).

We have:

$$\frac{2^{1-2\nu}}{\Gamma(\nu)} x^\nu K_\nu^2(x) = \frac{2^{1-2\nu}}{\Gamma(\nu)} x^\nu \underbrace{\frac{1}{2}\int_0^\infty \frac{1}{t} e^{-\frac{t}{2}-\frac{x^2}{t}} K_\nu\left(\frac{x^2}{t}\right) \mathrm{d}t}_{(3.7)}$$

$$
\begin{aligned}
&> \frac{2^{1-2\nu}}{\Gamma(\nu)} x^{\nu} \, \frac{1}{2} \int_0^\infty \frac{1}{t} e^{-\frac{t}{2}-\frac{x^2}{t}} \underbrace{2^{\nu-1}\Gamma(\nu) e^{-\frac{x^2}{t}} \left(\frac{x^2}{t}\right)^{-\nu}}_{(3.8)} \, \mathrm{d}t \\
&= 2^{-\nu} \, \frac{1}{2} x^{-\nu} \int_0^\infty \frac{1}{t^{-\nu+1}} e^{-\frac{t}{2}-\frac{2x^2}{t}} \, \mathrm{d}t \\
&= \frac{1}{2} x^{-\nu} \underbrace{\int_0^\infty \frac{1}{u^{-\nu+1}} e^{-u-\frac{x^2}{u}} \, \mathrm{d}u}_{\text{changing variables to } u := \frac{1}{2}t} \\
&= \underbrace{K_{-\nu}(2x)}_{(3.9)} = \underbrace{K_{\nu}(2x)}_{(3.10)} .
\end{aligned}
$$

Note that (3.8) does not hold for $\nu < \frac{1}{2}$, and in fact the Matérn kernel is not pixie for such ν. □

3.3 Convergence bounds

Let $\tilde{f}(x, y) := \tilde{s}(x, y) - k(x, y)$, and $\breve{f}(x, y) := \breve{s}(x, y) - k(x, y)$. We know that $\mathbb{E}\, f(x, y) = 0$ and have a closed form for $\operatorname{Var} f(x, y)$, but to better understand the error behavior across inputs, we wish to bound $\|f\|$ for various norms.

3.3.1 L_2 bound

If μ is a finite measure on $\mathcal{X} \times \mathcal{X}$ ($\mu(\mathcal{X}^2) < \infty$), the $L_2(\mathcal{X}^2, \mu)$ norm of f is

$$
\|f\|_\mu^2 := \int_{\mathcal{X}^2} f(x, y)^2 \, \mathrm{d}\mu(x, y). \tag{3.11}
$$

Proposition 3.4. *Let k be a continuous shift-invariant positive-definite function $k(x, y) = \underline{k}(\Delta)$ defined on $\mathcal{X} \subseteq \mathbb{R}^d$, with $\underline{k}(0) = 1$. Let μ be a finite measure on $\mathcal{X}^2$, and define $\|\cdot\|_\mu^2$ as in (3.11). Define $\tilde{z}$ as in (3.1) and let $\tilde{f}(x, y) := \tilde{z}(x)^\mathsf{T} \tilde{z}(y) - k(x, y)$. Then*

(i) The expected squared L_2 norm of the error is

$$
\mathbb{E}\|\tilde{f}\|_\mu^2 = \frac{1}{D} \int_{\mathcal{X}^2} \left[1 + k(2x, 2y) - 2k(x, y)^2\right] \mathrm{d}\mu(x, y).
$$

(ii) The L_2 norm of the error concentrates around its expectation at least exponentially:

$$
\Pr\left(\left|\|\tilde{f}\|_\mu^2 - \mathbb{E}\|\tilde{f}\|_\mu^2\right| \geq \varepsilon\right) \leq 2 \exp\left(\frac{-D^3 \varepsilon^2}{32(2D+1)^2 \mu(\mathcal{X}^2)^2}\right) \leq 2 \exp\left(\frac{-D\varepsilon^2}{288 \mu(\mathcal{X}^2)^2}\right).
$$

Proposition 3.5. *Let k, μ, and $\|\cdot\|_\mu$ be as in Proposition 3.4. Define $\breve{z}$ as in (3.2) and let $\breve{f}(x, y) = \breve{z}(x)^\mathsf{T} \breve{z}(y) - k(x, y)$. Then* *(i) The expected squared L_2 norm of the error is*

$$\mathbb{E}\|\breve{f}\|_\mu^2 = \frac{1}{D}\int_{\mathcal{X}^2}\left[1 + \frac{1}{2}k(2x, 2y) - k(x, y)^2\right] \mathrm{d}\mu(x, y).$$

(ii) The L_2 norm of the error concentrates around its expectation at least exponentially:

$$\Pr\left(\left|\|\breve{f}\|_\mu^2 - \mathbb{E}\|\breve{f}\|_\mu^2\right| \geq \varepsilon\right) \leq 2\exp\left(\frac{-D^3\varepsilon^2}{128(3D+2)^2\mu(\mathcal{X}^2)^2}\right) \leq 2\exp\left(\frac{-D\varepsilon^2}{3200\mu(\mathcal{X}^2)^2}\right).$$

The proofs for these propositions are simple applications of Tonelli's theorem and McDiarmid bounds; full details are given in Appendices B.1 and B.2.

Thus for the kernels considered above, the expected $L_2(\mu)$ error for $\tilde{z}$ is less than that of $\breve{z}$; the comparable concentration bound is also tighter. The second inequality is simpler, but somewhat looser for $D \gg 1$; asymptotically, the coefficient of the denominator would be 128 for $\tilde{f}$ (instead of 288) and 1152 for $\breve{f}$ (instead of 3200).

Note that if $\mu = \mu_X \times \mu_Y$ is a joint distribution of independent random variables, then

$$\begin{aligned}\mathbb{E}\|\tilde{f}\|_\mu^2 &= \frac{1}{D}\left[1 + \textsc{mmk}_k(\mu_{2X}, \mu_{2Y}) - 2\,\textsc{mmk}_{k^2}(\mu_X, \mu_y)\right]\\ \mathbb{E}\|\breve{f}\|_\mu^2 &= \frac{1}{D}\left[1 + \tfrac{1}{2}\,\textsc{mmk}_k(\mu_{2X}, \mu_{2Y}) - \textsc{mmk}_{k^2}(\mu_X, \mu_y)\right].\end{aligned}$$

Sriperumbudur and Szabó (2015, Corollary 2 and Theorem 3) subsequently bounded the deviation of f in the L_r norm for any $r \in [1, \infty)$, but only for μ the Lebesgue measure. Let ℓ be the diameter of $\mathcal{X}$ and C be some (unspecified) universal constant. Then their bound for L_2 amounts to, for ε large enough such that the term inside the parentheses is nonnegative,

$$\Pr\left(\|\tilde{f}\|_{L_2(\mathcal{X})} \geq \varepsilon\right) \leq \exp\left(-\frac{1}{2}\left(\frac{2^d\Gamma\left(\frac{d}{2}+1\right)}{\pi^{d/2}\ell^d}\sqrt{\frac{D}{2}}\varepsilon - C\right)^2\right).$$

This has the same asymptotic rate in terms of D and ε as our bound but, since $\mu(\mathcal{X}^2) = O(\ell^{2d})$, has better dependence on ℓ.

3.3.2 High-probability uniform bound

Claim 1 of Rahimi and Recht (2007) is that if $\mathcal{X} \subset \mathbb{R}^d$ is compact with diameter ℓ,[2]

$$\Pr\left(\|f\|_\infty \geq \varepsilon\right) \leq 256\left(\frac{\sigma_p \ell}{\varepsilon}\right)^2 \exp\left(-\frac{D\varepsilon^2}{8(d+2)}\right),$$

where $\sigma_p^2 = \mathbb{E}[\omega^\mathsf{T}\omega] = \operatorname{tr}\nabla^2\underline{k}(0)$ depends on the kernel.

It was not necessarily clear in that paper that the bound applies only to $\tilde{s}$ and not $\breve{s}$; we can also tighten some constants. We first state the tightened bound for $\tilde{z}$.

[2] Note our D is half that in Rahimi and Recht (2007), since we want to compare embeddings of the same dimension.

Proposition 3.6. *Let k be a continuous shift-invariant positive-definite function $k(x, y) = \underline{k}(\Delta)$ defined on $\mathcal{X} \subset \mathbb{R}^d$, with $\underline{k}(0) = 1$ and such that $\nabla^2 \underline{k}(0)$ exists. Suppose $\mathcal{X}$ is compact, with diameter ℓ. Denote $\underline{k}$'s Fourier transform as $\Omega(\omega)$, which will be a probability distribution due to Bochner's theorem; let $\sigma_p^2 = \mathbb{E}_p \|\omega\|^2$. Let $\tilde{z}$ be as in (3.1), and define $\tilde{f}(x, y) := \tilde{z}(x)^\mathsf{T} \tilde{z}(y) - k(x, y)$. For any $\varepsilon > 0$, let*

$$\alpha_\varepsilon := \min\left(1, \sup_{x,y\in\mathcal{X}} \frac{1}{2} + \frac{1}{2} k(2x, 2y) - k(x, y)^2 + \tfrac{1}{6}\varepsilon\right), \qquad \beta_d := \left(\left(\tfrac{d}{2}\right)^{\frac{-d}{d+2}} + \left(\tfrac{d}{2}\right)^{\frac{2}{d+2}}\right) 2^{\frac{6d+2}{d+2}}.$$

Then

$$\begin{aligned}\Pr\left(\|\tilde{f}\|_\infty \geq \varepsilon\right) &\leq \beta_d \left(\frac{\sigma_p \ell}{\varepsilon}\right)^{\frac{2}{1+\frac{2}{d}}} \exp\left(-\frac{D\varepsilon^2}{8(d+2)\alpha_\varepsilon}\right) \\ &\leq 66 \left(\frac{\sigma_p \ell}{\varepsilon}\right)^2 \exp\left(-\frac{D\varepsilon^2}{8(d+2)}\right) \quad \text{if } \varepsilon \leq \sigma_p \ell.\end{aligned}$$

Thus, we can achieve an embedding with pointwise error no more than ε with probability at least $1 - \delta$ as long as

$$D \geq \frac{8(d+2)\alpha_\varepsilon}{\varepsilon^2} \left[\frac{2}{1 + \frac{2}{d}} \log \frac{\sigma_p \ell}{\varepsilon} + \log \frac{\beta_d}{\delta}\right].$$

The proof strategy is very similar to that of Rahimi and Recht (2007): place an ε-net with radius r over $\mathcal{X}_\Delta := \{x - y : x, y \in \mathcal{X}\}$, bound the error $\tilde{f}$ by $\varepsilon/2$ at the centers of the net by Hoeffding's inequality (1963), and bound the Lipschitz constant of $\tilde{f}$, which is at most that of $\tilde{s}$, by $\varepsilon/(2r)$ with Markov's inequality. The introduction of α_ε is by replacing Hoeffding's inequality with that of S. Bernstein (1924) when it is tighter, using the variance from (3.4). The constant β_d is obtained by exactly optimizing the value of r, rather than the algebraically simpler value originally used; $\beta_{64} = 66$ is its maximum, and $\lim_{d\to\infty} \beta_d = 64$, though it is lower for small d, as shown in Figure 3.2. The additional hypothesis, that $\nabla^2 k(0)$ exists, is equivalent to the existence of the first two moments of $P(\omega)$; a finite first moment is used in the proof, and of course without a finite second moment the bound is vacuous. The full proof is given in Appendix B.3.

For any pixie kernel, $\alpha_\varepsilon \leq \frac{1}{2} + \frac{1}{6}\varepsilon$; the Bernstein bound is tighter at least when $\varepsilon < 3$. (Recall that the maximal possible error is $\varepsilon = 2$, so it is essentially always preferable.) For the Gaussian kernel of bandwidth σ, $\sigma_p^2 = d/\sigma^2$.

For $\breve{z}$, since the embedding $\breve{s}$ is not shift-invariant, we must instead place the ε-net on $\mathcal{X}^2$. The additional noise in $\breve{s}$ also increases the expected Lipschitz constant and gives looser bounds on each term in the sum, though there are twice as many such terms. The corresponding bound is as follows:

Proposition 3.7. *Let k, $\mathcal{X}$, ℓ, $\Omega(\omega)$, and σ_p be as in Proposition 3.6. Define $\breve{z}$ by (3.2), and $\breve{f}(x, y) := \breve{z}(x)^\mathsf{T} \breve{z}(y) - k(x, y)$. For any $\varepsilon > 0$, define*

$$\alpha'_\varepsilon := \min\left(1, \sup_{x,y\in\mathcal{X}} \tfrac{1}{4} + \tfrac{1}{8} k(2x, 2y) - \tfrac{1}{4} k(x, y)^2 + \tfrac{1}{12}\varepsilon\right), \qquad \beta'_d := \left(d^{\frac{-d}{d+1}} + d^{\frac{1}{d+1}}\right) 2^{\frac{5d+1}{d+1}} 3^{\frac{d}{d+1}}.$$

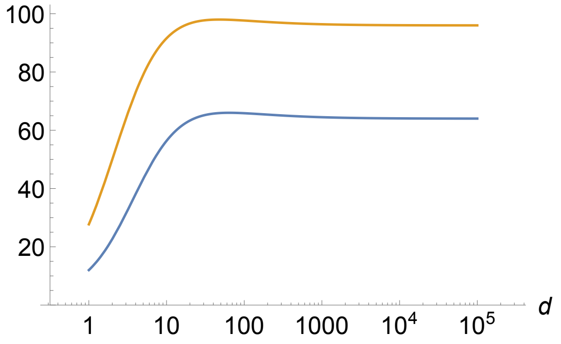


Figure 3.2: The coefficient β_d of Proposition 3.6 (blue, for $\tilde{z}$) and β'_d of Proposition 3.7 (orange, for $\breve{z}$). Rahimi and Recht (2007) used a constant of 256 for $\tilde{z}$.

Then

$$\begin{aligned}\Pr\left(\|\breve{f}\|_\infty \geq \varepsilon\right) &\leq \beta'_d \left(\frac{\sigma_p \ell}{\varepsilon}\right)^{\frac{2}{1+\frac{1}{d}}} \exp\left(-\frac{D\varepsilon^2}{32(d+1)\alpha'_\varepsilon}\right) \\ &\leq 98 \left(\frac{\sigma_p \ell}{\varepsilon}\right)^2 \exp\left(-\frac{D\varepsilon^2}{32(d+1)}\right) \qquad \textit{when } \varepsilon \leq \sigma_p \ell.\end{aligned}$$

Thus, we can achieve an embedding with pointwise error no more than ε with probability at least $1-\delta$ as long as

$$D \geq \frac{32(d+1)\alpha'_\varepsilon}{\varepsilon^2}\left[\frac{2}{1+\frac{1}{d}} \log \frac{\sigma_p \ell}{\varepsilon} + \log \frac{\beta'_d}{\delta}\right].$$

$\beta'_{48} = 98$, and $\lim_{d\to\infty} \beta'_d = 96$, also shown in Figure 3.2. The full proof is given in Appendix B.4.

For *any* kernel, pixie or not, the Bernstein bound is superior for any $\varepsilon < 7.5$.

Note that when the Bernstein bound is being used for a typical pixie kernel, $\alpha_\varepsilon \approx 2\alpha'_\varepsilon$.

Although we cannot use these bounds to conclude that $\|\tilde{f}\|_\infty < \|\breve{f}\|_\infty$, the fact that $\tilde{f}$ yields smaller bounds using the same techniques certainly suggests that it might be usually true.

3.3.3 Expected max error

Noting that $\mathbb{E}\|f\|_\infty = \int_0^\infty \Pr\left(\|f\|_\infty \geq \varepsilon\right) \mathrm{d}\varepsilon$, one could consider bounding $\mathbb{E}\|f\|_\infty$ via Propositions 3.6 and 3.7. Unfortunately, that integral diverges on $(0,\gamma)$ for any $\gamma > 0$. If we instead integrate the minimum of that bound and 1, the result depends on a solution to a transcendental equation, so analytical manipulation is difficult.

We can, however, use a slight generalization of Dudley's entropy integral (1967) to obtain the following bound:

Proposition 3.8. *Let k, $\mathcal{X}$, ℓ, and $\Omega(\omega)$ be as in Proposition 3.6. Define $\tilde{z}$ by (3.1), and $\tilde{f}(x,y) := \tilde{z}(x)^\mathsf{T}\tilde{z}(y) - k(x,y)$. Let $\mathcal{X}_\Delta := \{x-y \mid x,y \in \mathcal{X}\}$; suppose k is L-Lipschitz on $\mathcal{X}_\Delta$. Let*

$R := \mathbb{E} \max_{i=1,\dots,\frac{D}{2}} \|\omega_i\|$. *Then*

$$\mathbb{E}\left[\|\tilde{f}\|_\infty\right] \le \frac{24\gamma\sqrt{d}\ell}{\sqrt{D}}(R+L)$$

where $\gamma \approx 0.964$.

The proof is given in Appendix B.5. In order to apply the method of Dudley (1967), we must work around $\|\omega_i\|$ (which appears in the covariance of the error process) being potentially unbounded. To do so, we bound a process with truncated $\|\omega_i\|$, and then relate that bound to $\tilde{f}$.

For the Gaussian kernel, $L = 1/(\sigma\sqrt{e})$ and[3]

$$R \le \left(\sqrt{2}\,\frac{\Gamma\left((d+1)/2\right)}{\Gamma\left(d/2\right)} + \sqrt{2\log\left(D/2\right)}\right)/\sigma \le \left(\sqrt{d} + \sqrt{2\log\left(D/2\right)}\right)/\sigma.$$

Thus

$$\mathbb{E}\|\tilde{f}\|_\infty < \frac{24\gamma\sqrt{d}\,\ell}{\sqrt{D}\,\sigma}\left(e^{-1/2} + \sqrt{d} + \sqrt{2\log(D/2)}\right). \tag{3.12}$$

Analagously, for the $\breve{z}$ features:

Proposition 3.9. *Let* $k, \mathcal{X}, \ell$, *and* $\Omega(\omega)$ *be as in Proposition 3.6. Define* $\breve{z}$ *by (3.2), and* $\breve{f}(x,y) := \breve{z}(x)^\mathsf{T}\breve{z}(y) - k(x,y)$. *Suppose* $k(\Delta)$ *is* L*-Lipschitz. Let* $R' := \mathbb{E}\max_{i=1,\dots,D}\|\omega_i\|$. *Then, for* $\mathcal{X}$ *and* D *not extremely small,*

$$\mathbb{E}\left[\|\breve{f}\|_\infty\right] \le \frac{48\gamma'_\mathcal{X}\ell\sqrt{d}}{\sqrt{D}}(R'+L)$$

where $0.803 < \gamma'_\mathcal{X} < 1.542$. *See Appendix B.6 for details on* $\gamma'_\mathcal{X}$ *and the "not extremely small" assumption.*

The proof is given in Appendix B.6. It is similar to that for Proposition 3.8, but the lack of shift invariance increases some constants and otherwise slightly complicates matters. Note also that the R' of Proposition 3.9 is slightly larger than the R of Proposition 3.8.

These two bounds are both quite loose in practice.

3.3.4 Concentration about the mean

Bousquet's inequality (2002) can be used to show exponential concentration of $\sup f$ about its mean.

We consider $\tilde{f}$ first. Let

$$\tilde{f}_\omega(\Delta) := \frac{2}{D}\left(\cos(\omega^\mathsf{T}\Delta) - \underline{k}(\Delta)\right),$$

[3] By the Gaussian concentration inequality (Boucheron et al. 2013, Theorem 5.6), each $\|\omega\| - \mathbb{E}\|\omega\|$ is sub-Gaussian with variance factor σ^{-2}; the claim follows from their Section 2.5.

so $\tilde{f}(\Delta) = \sum_{i=1}^{D/2} \tilde{f}_{\omega_i}(\Delta)$. Define the "wimpy variance" of $\tilde{f}/2$ (which we use so that $|\tilde{f}/2| \le 1$) as

$$\begin{aligned}
\sigma^2_{\tilde{f}/2} &:= \sup_{\Delta \in \mathcal{X}_\Delta} \sum_{i=1}^{D/2} \operatorname{Var}\left[\tfrac{1}{2}\tilde{f}_{\omega_i}(\Delta)\right] \\
&= \sup_{\Delta \in \mathcal{X}_\Delta} \frac{1}{2D} \operatorname{Var}\left[\cos(\omega^\mathsf{T}\Delta)\right] \\
&= \frac{1}{4D} \sup_{\Delta \in \mathcal{X}_\Delta} \left[1 + \underline{k}(2\Delta) - 2\underline{k}(\Delta)^2\right] \\
&=: \frac{1}{4D}\sigma_w^2.
\end{aligned}$$

Clearly $0 \le \sigma_w^2 \le 2$; for pixie kernels, $\sigma_w^2 \le 1$, with it approaching unity for typical kernels on domains large relative to the lengthscale.

Proposition 3.10. *Let k, $\mathcal{X}$, and $\Omega(\omega)$ be as in Proposition 3.6, and $\tilde{z}$ be defined by (3.1). Let $\tilde{f}(\Delta) = \tilde{z}(x)^\mathsf{T}\tilde{z}(y) - \underline{k}(\Delta)$ for $\Delta = x - y$, and $\sigma_w^2 := \sup_{\Delta \in \mathcal{X}_\Delta} 1 + \underline{k}(2\Delta) - 2\underline{k}(\Delta)^2$. Then*

$$\Pr\left(\|\tilde{f}\|_\infty - \mathbb{E}\|\tilde{f}\|_\infty \ge \varepsilon\right) \le 2\exp\left(-\frac{D\varepsilon^2}{8D\ \mathbb{E}\|\tilde{f}\|_\infty + 2\sigma_w^2 + \frac{4}{3}D\varepsilon}\right).$$

Proof. We use the Bernstein-style form of Theorem 12.5 of Boucheron et al. (2013) on $\tilde{f}(\Delta)/2$ to obtain that

$$\begin{aligned}
\Pr\left(\sup\frac{\tilde{f}}{2} - \mathbb{E}\sup\frac{\tilde{f}}{2} \ge t\right) &\le \exp\left(-\frac{t^2}{4\,\mathbb{E}\sup\frac{\tilde{f}}{2} + 2\sigma^2_{\tilde{f}/2} + \frac{2}{3}t}\right) \\
\Pr\left(\sup\tilde{f} - \mathbb{E}\sup\tilde{f} \ge \varepsilon\right) &\le \exp\left(-\frac{\frac{1}{4}\varepsilon^2}{2\,\mathbb{E}\sup\tilde{f} + \frac{1}{2D}\sigma_w^2 + \frac{1}{3}\varepsilon}\right) \\
&= \exp\left(-\frac{D\varepsilon^2}{8D\,\mathbb{E}\sup\tilde{f} + 2\sigma_w^2 + \frac{4}{3}D\varepsilon}\right).
\end{aligned}$$

The same holds for $-\tilde{f}$, and $\mathbb{E}\sup\tilde{f} \le \mathbb{E}\|f\|_\infty$, $\mathbb{E}\sup(-\tilde{f}) \le \mathbb{E}\|f\|_\infty$. The claim follows by a union bound. □

A bound on the lower tail, unfortunately, is not available in the same form.

For $\breve{f}$, note $|\breve{f}| \le 3$, so we use $\breve{f}/3$. Letting $\breve{f}_{\omega,b}(x, y) := \frac{1}{D}(\cos(\omega^\mathsf{T}(x - y)) + \cos(\omega^\mathsf{T}(x + y) + 2b) - k(x, y))$, we have

$$\begin{aligned}
\sigma^2_{\breve{f}/3} &: = \sup_{x,y \in \mathcal{X}} \sum_{i=1}^{D} \operatorname{Var}\left[\tfrac{1}{3}\breve{f}_{\omega_i,b_i}(\Delta)\right] \\
&= \sup_{x,y \in \mathcal{X}} \frac{1}{9D}\left[1 + \tfrac{1}{2}k(2\Delta) - k(\Delta)^2\right] \\
&= \frac{1}{18D}(1 + \sigma_w^2),
\end{aligned}$$

Thus the same argument gives us:

Proposition 3.11. *Let k, $\mathcal{X}$, and $\Omega(\omega)$ be as in Proposition 3.6. Let $\breve{z}$ be as in (3.2), $\tilde{f}(x, y) := \tilde{z}(x)^\mathsf{T}\tilde{z}(y) - k(x, y)$, and define σ_w as in Proposition 3.10. Then*

$$\Pr\left(\|\breve{f}\|_\infty - \mathbb{E}\|\breve{f}\|_\infty \geq \varepsilon\right) \leq 2\exp\left(-\frac{D\varepsilon^2}{12D\ \mathbb{E}\|\breve{f}\|_\infty + \frac{1}{2}(1 + \sigma_w^2) + 2D\varepsilon}\right).$$

Note that the bound for $\tilde{f}$ strictly dominates the bound for $\breve{f}$ only in the unlikely case of $\sigma_w^2 < \frac{1}{3}$.

3.3.5 Other bounds

Sriperumbudur and Szabó (2015) later proved a rate-optimal $O_P(D^{-1/2})$ bound on $\|\tilde{f}\|_\infty$. Phrased in the terminology we use here, it amounts to:

$$\Pr\left(\|\tilde{f}\|_\infty \geq \varepsilon\right) = \begin{cases} 1 & \varepsilon < \frac{h}{\sqrt{D/2}} \\ \exp\left(-\frac{1}{2}\left(\sqrt{\frac{D}{2}}\varepsilon - h\right)^2\right) & \text{otherwise} \end{cases}$$

$$\text{where} \quad h := 32\sqrt{2d\log(\ell + 1)} + 32\sqrt{2d\log(\sigma_p + 1)} + 16\sqrt{\frac{2d}{\log(\ell + 1)}}$$

In practice, for moderately-sized inputs, the constants can be much worse than the non-optimal bound of Proposition 3.6. For example, the regime of Figure 3.5 is $d = 1$, $\ell = 6$, $\sigma_p = 1$. In that setting, the smallest D for which even $\Pr\left(\|\tilde{f}\|_\infty \geq 1\right)$ can be shown to be less than unity is a staggering $D = 27\,392$, compared to the 500 plotted for the other bounds.

3.4 Downstream error

When we use random Fourier features, the final output of our analysis is not simply estimates of the values of the kernel function; rather, we wish to use this kernel within some machine learning framework. A natural question, then, is: how much does the use of a random Fourier features approximation change the outcome of the prediction compared to if we had used the exact kernel?

One approach to answering this question is to study the difference between functions in the original kernel RKHS versus functions in the RKHS corresponding to the approximation. This is the approach taken by Rahimi and Recht (2008a,b), as well as the later work of Bach (2015) and Rudi et al. (2016). Rudi et al. (2016), in particular, provide an invaluable theoretical study of the effect of using random features in regression models.

In some contexts, however, we would prefer to consider not the learning-theoretic convergence of hypotheses to the assumed "true" function, but rather directly consider the difference in predictions due to using the z embedding instead of the exact kernel k. We give a few such bounds here. We stress, however, that combining these results with standard learning rates for the models yields worse bounds compared to those of Bach (2015) and Rudi et al. (2016).

3.4.1 Kernel ridge regression

We first consider kernel ridge regression (KRR; Saunders et al. 1998). Suppose we are given n training pairs $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$ as well as a regularization parameter $\lambda = n\lambda_0 > 0$. We construct the training Gram matrix K by $K_{ij} = k(x_i, x_j)$. KRR gives predictions $h(x) = \alpha^\mathsf{T} k_x$, where $\alpha = (K + \lambda I)^{-1} y$ and k_x is the vector with ith component $k(x_i, x)$.[4] When using Fourier features, one would not use α, but instead a primal weight vector w; still, it will be useful for us to analyze the situation in the dual.

Proposition 1 of Cortes et al. (2010) bounds the change in KRR predictions from approximating the kernel matrix K by $\hat{K}$, in terms of $\|\hat{K} - K\|_2$. They assume, however, that the kernel evaluations at test time k_x are unapproximated, which is certainly not the case when using Fourier features. We therefore extend their result to Proposition 3.12 before using it to analyze the performance of Fourier features.

Proposition 3.12. *Given a training set $\{(x_i, y_i)\}_{i=1}^n$, with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, let $h(x)$ denote the result of kernel ridge regression using the PSD training kernel matrix K and test kernel values k_x. Let $\hat{h}(x)$ be the same using a PSD approximation to the training kernel matrix $\hat{K}$ and test kernel values $\hat{k}_x$. Further assume that the training labels are centered, $\sum_{i=1}^n y_i = 0$, and let $\sigma_y^2 := \frac{1}{n} \sum_{i=1}^n y_i^2$. Also suppose $\|k_x\|_\infty \le \kappa$. Then:*

$$|h'(x) - h(x)| \le \frac{\sigma_y}{\sqrt{n}\lambda_0} \|\hat{k}_x - k_x\| + \frac{\kappa \sigma_y}{n \lambda_0^2} \|\hat{K} - K\|_2.$$

Proof. Let $\alpha = (K + \lambda I)^{-1} y$, $\hat{\alpha} = (\hat{K} + \lambda I)^{-1} y$. Thus, using $\hat{M}^{-1} - M^{-1} = -\hat{M}^{-1}(\hat{M} - M) M^{-1}$, we have

$$\begin{aligned}
\hat{\alpha} - \alpha &= -(\hat{K} + \lambda I)^{-1} (\hat{K} - K)(K + \lambda I)^{-1} y \\
\|\hat{\alpha} - \alpha\| &\le \|(\hat{K} + \lambda I)^{-1}\|_2 \, \|\hat{K} - K\|_2 \, \|(K + \lambda I)^{-1}\|_2 \, \|y\| \\
&\le \frac{1}{\lambda^2} \|\hat{K} - K\|_2 \, \|y\|
\end{aligned}$$

since the smallest eigenvalues of $K + \lambda I$ and $\hat{K} + \lambda I$ are at least λ. Since $\|k_x\| \le \sqrt{n}\kappa$ and $\|\hat{\alpha}\| \le \|y\| / \lambda$:

$$\begin{aligned}
|\hat{h}(x) - h(x)| &= |\hat{\alpha}^\mathsf{T} \hat{k}_x - \alpha^\mathsf{T} k_x| \\
&= |\hat{\alpha}^\mathsf{T} (\hat{k}_x - k_x) + (\hat{\alpha} - \alpha)^\mathsf{T} k_x| \\
&\le \|\hat{\alpha}\| \|\hat{k}_x - k_x\| + \|\hat{\alpha} - \alpha\| \|k_x\| \\
&\le \frac{\|y\|}{\lambda} \|\hat{k}_x - k_x\| + \frac{\sqrt{n}\kappa \|y\|}{\lambda^2} \|\hat{K} - K\|_2.
\end{aligned}$$

The claim follows from $\lambda = n\lambda_0$, $\|y\| = \sqrt{n}\sigma_y$. □

[4] If a bias term is desired, we can use $k'(x, x') = k(x, x') + 1$ by appending a constant feature 1 to the embedding z. Because this change is accounted for exactly, it affects the error analysis here only in that we must use $\sup |k(x, y)| \le 2$, in which case the first factor of (3.13) becomes $(\lambda_0 + 2) / \lambda_0^2$.

Suppose that, per the uniform error bounds of Section 3.3.2, $\sup |k(x, y) - s(x, y)| \le \varepsilon$. Then $\|\hat{k}_x - k_x\| \le \sqrt{n}\varepsilon$ and $\|\hat{K} - K\|_2 \le \|\hat{K} - K\|_F \le n\varepsilon$, and Proposition 3.12 gives

$$\left|\hat{h}(x) - h(x)\right| \le \frac{\sigma_y}{\sqrt{n}\lambda_0}\sqrt{n}\varepsilon + \frac{\sigma_y}{n\lambda_0^2} n\varepsilon \le \frac{\lambda_0 + 1}{\lambda_0^2}\sigma_y \varepsilon. \tag{3.13}$$

Thus

$$\Pr\left(|h'(x) - h(x)| \ge \varepsilon\right) \le \Pr\left(\|f\|_\infty \ge \frac{\lambda_0^2 \varepsilon}{(\lambda_0 + 1)\sigma_y}\right).$$

which we can bound with Proposition 3.6 or 3.7. We can therefore guarantee $|h(x) - h'(x)| \le \varepsilon$ with probability at least δ if

$$D = \Omega\left(d \left(\frac{(\lambda_0 + 1)\sigma_y}{\lambda_0^2 \, \varepsilon}\right)^2 \left[\log \frac{1}{\delta} + \log \frac{(\lambda_0 + 1)\sigma_y}{\lambda_0^2 \varepsilon} + \log \sigma_p \ell\right]\right).$$

Note that this rate does not depend on n.

If we want $|h'(x) - h(x)| = O\left(\frac{1}{\sqrt{n}}\right)$ in order to match $h(x)$'s convergence rate (Bousquet and Elisseeff 2001), ignoring the logarithmic terms, we thus need $D = \Omega(n)$, matching the conclusion of Rahimi and Recht (2008a). It is worth saying again, however, that Bach (2015) and Rudi et al. (2016) obtained better rates depending on the form of the particular learning problem.

3.4.2 Support vector machines

We will now give a similar bound for SVM classifiers. We will see that this method gives much worse results than in the ridge regression case; RKHS analyses should be used here instead.

Consider an SVM classifier with no offset, such that $h(x) = w^\mathsf{T}\Phi(x)$ for a kernel embedding $\Phi(x) : \mathcal{X} \to \mathcal{H}$ and w is found by

$$\operatorname*{argmin}_{w \in \mathcal{H}} \frac{1}{2}\|w\|^2 + \frac{C_0}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i \langle w, \Phi(x_i)\rangle\right)$$

where $\{(x_i, y_i)\}_{i=1}^n$ is our training set with $y_i \in \{-1, 1\}$, and the decision function is $h(x) = \langle w, \Phi(x)\rangle$.[5] For a given x, Cortes et al. (2010) consider an embedding in $\mathcal{H} = \mathbb{R}^{n+1}$ which is equivalent on the given set of points. They bound $\left|\hat{h}(x) - h(x)\right|$ in terms of $\|\hat{K} - K\|_2$ in their Proposition 2, but again assume that the test-time kernel values k_x are exact. We will again extend their result in Proposition 3.13:

Proposition 3.13. *Given a training set $\{(x_i, y_i)\}_{i=1}^n$, with $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$, let $h(x)$ denote the decision function of an SVM classifier using the PSD training matrix K and test kernel values k_x. Let $\hat{h}(x)$ be the same using a PSD approximation to the training kernel matrix $\hat{K}$ and test kernel values $\hat{k}_x$. Suppose $\sup_x k(x, x) \le \kappa$. Define $\delta_x := \|\hat{K} - K\|_2 + \|\hat{k}_x - k_x\| + |\hat{k}(x, x) - k(x, x)|$. Then:*

$$|\hat{h}(x) - h(x)| \le \sqrt{2}\,\kappa^{\frac{3}{4}}\, C_0\, \delta_x^{1/4} + \sqrt{\kappa}\, C_0\, \delta_x^{1/2}.$$

[5] We again assume there is no bias term for simplicity; adding a constant feature again changes the analysis only in that it makes the κ of Proposition 3.13 2 instead of 1.

Proof. Use the setup of Section 2.2 of Cortes et al. (2010). In particular, we will use $\|w\| \leq \sqrt{\kappa} C_0$ and their (16-17):

$$\Phi(x_i) = K_x^{1/2} e_i$$
$$\|\hat{w} - w\|^2 \leq 2C_0^2 \sqrt{\kappa} \|\hat{K}_x^{1/2} - K_x^{1/2}\|,$$

where $K_x := \begin{bmatrix} K & k_x \\ k_x^\mathsf{T} & k(x,x) \end{bmatrix}$ and e_i is the ith standard basis. Also let $f_x := \hat{k}(x,x) - k(x,x)$.

Further, Lemma 1 of Cortes et al. (2010) says that $\|\hat{K}_x^{1/2} - K_x^{1/2}\|_2 \leq \|\hat{K}_x - K_x\|_2^{1/2}$. Let $f_x := \hat{k}(x,x) - k(x,x)$; then, by Weyl's inequality for singular values,

$$\left\| \begin{bmatrix} \hat{K} - K & \hat{k}_x - k_x \\ \hat{k}_x^\mathsf{T} - k_x^\mathsf{T} & f_x \end{bmatrix} \right\|_2 \leq \|\hat{K} - K\|_2 + \|\hat{k}_x - k_x\| + |f_x| \,.$$

Thus

$$\begin{aligned}
&|\hat{h}(x) - h(x)| \\
&\quad = \left| (\hat{w} - w)^\mathsf{T} \hat{\Phi}(x) + w^\mathsf{T} (\hat{\Phi}(x) - \Phi(x)) \right| \\
&\quad \leq \|\hat{w} - w\| \|\hat{\Phi}(x)\| + \|w\| \|\hat{\Phi}(x) - \Phi(x)\| \\
&\quad \leq \sqrt{2} \kappa^{\frac{1}{4}} C_0 \|\hat{K}_x^{1/2} - K_x^{1/2}\|_2^{1/2} \sqrt{\kappa} + \sqrt{\kappa} C_0 \|(\hat{K}_x^{1/2} - K_x^{1/2}) e_{n+1}\| \\
&\quad \leq \sqrt{2} \kappa^{\frac{3}{4}} C_0 \|\hat{K}_x - K_x\|_2^{1/4} + \sqrt{\kappa} C_0 \|\hat{K}_x - K_x\|^{1/2} \\
&\quad \leq \sqrt{2} \kappa^{\frac{3}{4}} C_0 \left(\|\hat{K} - K\|_2 + \|\hat{k}_x - k_x\| + |f_x| \right)^{1/4} + \sqrt{\kappa} C_0 \left(\|\hat{K} - K\|_2 + \|\hat{k}_x - k_x\| + |f_x| \right)^{1/2}
\end{aligned}$$

as claimed. □

Suppose that $\sup |k(x,y) - s(x,y)| \leq \varepsilon$. Then, as in the last section, $\|\hat{k}_x - k_x\| \leq \sqrt{n} \varepsilon$ and $\|\hat{K} - K\|_2 \leq n\varepsilon$. Then, letting γ be 0 for $\tilde{z}$ and 1 for $\breve{z}$, Proposition 3.13 gives

$$|\hat{h}(x) - h(x)| \leq \sqrt{2} C_0 \left(n + \sqrt{n} + \gamma \right)^{1/4} \varepsilon^{1/4} + C_0 \left(n + \sqrt{n} + \gamma \right)^{1/2} \varepsilon^{1/2}.$$

Then $|\hat{h}(x) - h(x)| \geq u$ only if

$$\varepsilon \leq \frac{2C_0^2 + 4C_0 u + u^2 - 2(C_0 + u)\sqrt{C_0(C_0 + 2u)}}{C_0^2 (n + \sqrt{n} + \gamma)}.$$

This bound has the unfortunate property of requiring the approximation to be *more* accurate as the training set size increases, and thus can prove only a very loose upper bound on the number of features needed to achieve a given approximation accuracy, due to the looseness of Proposition 3.13. Analyses of generalization error in the induced RKHS, such as Rahimi and Recht (2008a), T. Yang et al. (2012), and Bach (2015), are more useful in this case.

3.5 Numerical evaluation on an interval

We will conduct a detailed study of the approximations on the interval $\mathcal{X} = [-b, b]$. Specifically, we evenly spaced 1 000 points on $[-5, 5]$ and approximated the kernel matrix using both embeddings at $D \in \{50, 100, 200, \dots, 900, 1\,000, 2\,000, \dots, 9\,000, 10\,000\}$, repeating each trial 1 000 times, estimating $\|f\|_\infty$ and $\|f\|_\mu$ at those points. We do not consider $d > 1$, because obtaining a reliable estimate of $\sup|f|$ becomes very computationally expensive even for $d = 2$.

Figure 3.3 shows the behavior of $\mathbb{E}\|f\|_\infty$ as b increases for various values of D. As expected, the $\tilde{z}$ embeddings have almost no error near 0. The error increases out to one or two bandwidths, after which the curve appears approximately linear in ℓ/σ, as predicted by Propositions 3.8 and 3.9.

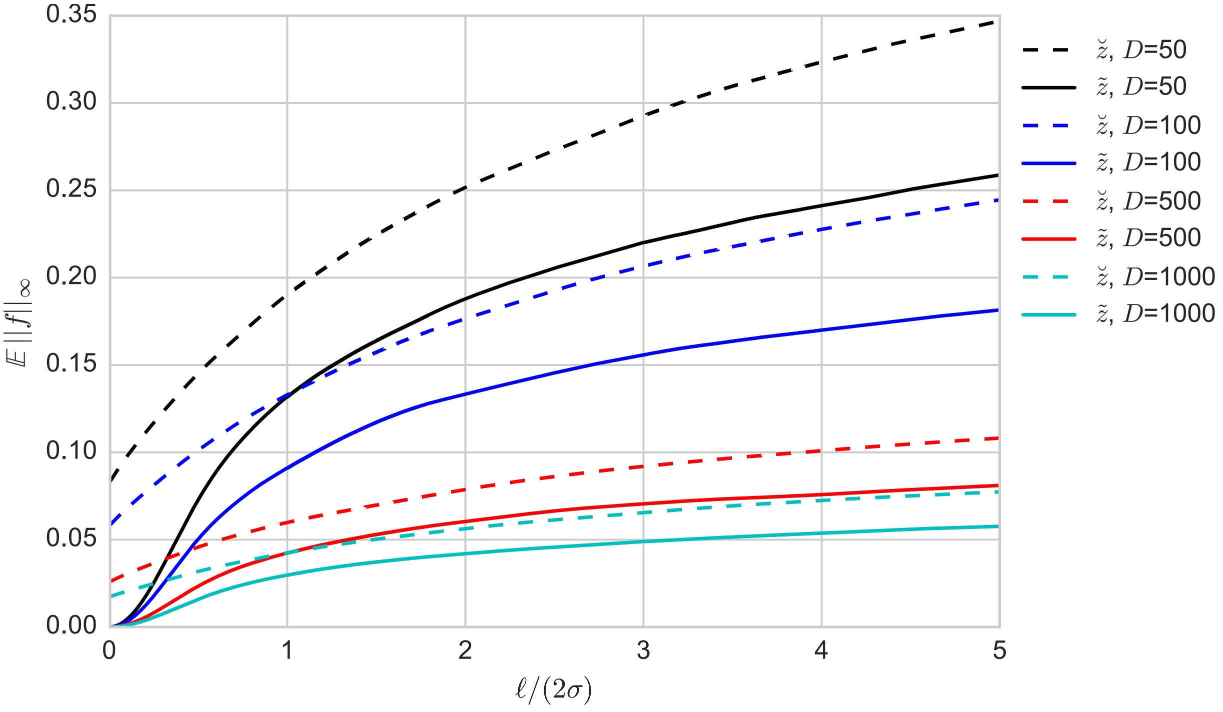


Figure 3.3: The maximum error within a given radius in $\mathbb{R}$, averaged over 1 000 evaluations.

Figure 3.4 fixes $b = 3$ and shows the expected maximal error as a function of D. It also plots the expected error obtained by numerically integrating the bounds of Propositions 3.6 and 3.7 (using the minimum of 1 and the stated bound). We can see that all of the bounds are fairly loose, but that the first version of the bound in the propositions (with β_d, the exponent depending on d, and α_ε) is substantially tighter than the second version when $d = 1$.

The bounds on $\mathbb{E}\|f\|_\infty$ of Propositions 3.8 and 3.9 are unfortunately too loose to show on the same plot. However, one important property does hold. For a fixed $\mathcal{X}$ and k, (3.12) predicts that $\mathbb{E}\|f\|_\infty = O(1/\sqrt{D})$. This holds empirically: performing linear regression of $\log \mathbb{E}\|\tilde{f}\|_\infty$ against $\log D$ yields a model of $\mathbb{E}\|\tilde{f}\|_\infty = e^c D^m$, with a 95% confidence interval for m of $[-0.502, -0.496]$; $\|\breve{f}\|_\infty$ gives $[-0.503, -0.497]$. The integrated bounds of Propositions 3.6 and 3.7 do not fit the scaling as a function of D nearly as well.

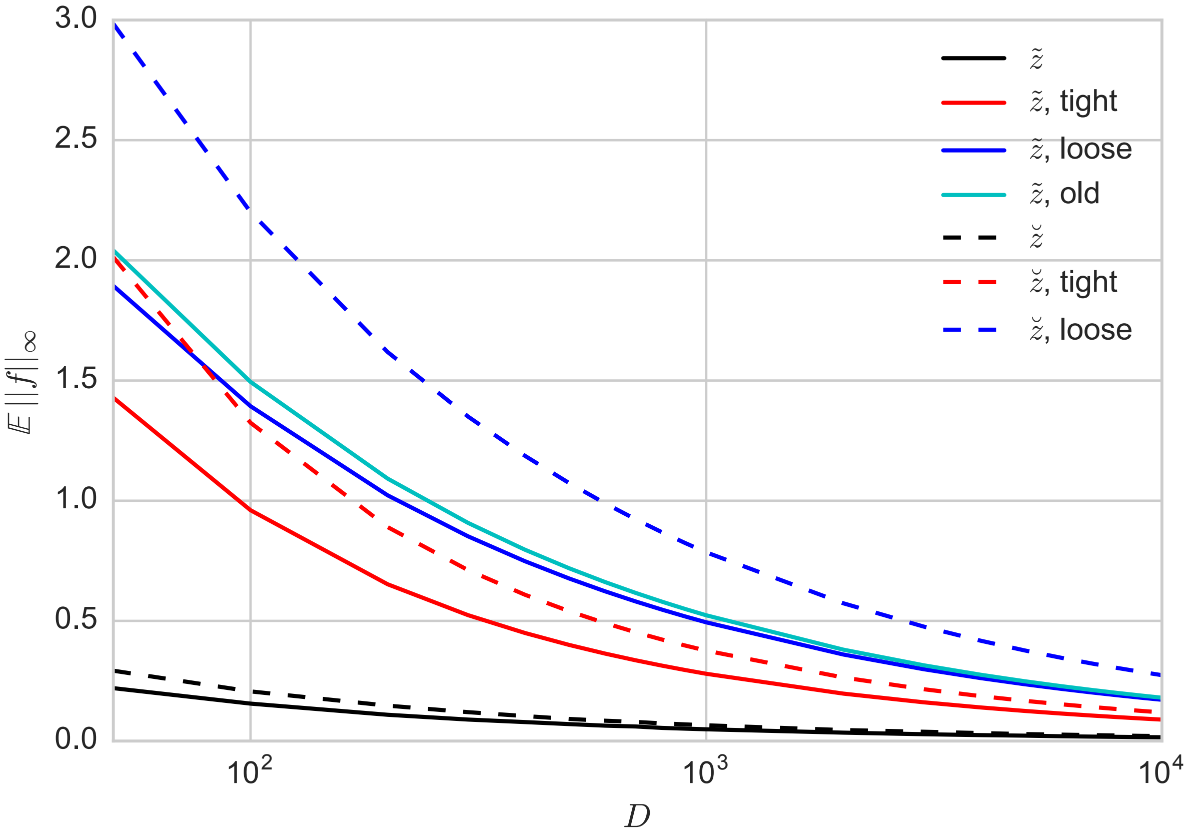


Figure 3.4: $\mathbb{E}\|f\|_\infty$ for the Gaussian kernel on $[-3, 3]$ with $\sigma = 1$, based on the mean of 1 000 evaluations and on numerical integration of the bounds from Propositions 3.6 and 3.7. ("Tight" refers to the bound with constants depending on d, and "loose" the second version; "old" is the version from Rahimi and Recht (2007).)

Figure 3.5 shows the empirical survival function of the max error for $D = 500$, along with the bounds of Propositions 3.6 and 3.7 and those of Propositions 3.10 and 3.11 using the empirical mean. The latter bounds are tighter than the former for low ε, especially for low D, but have a lower slope.

The mean of the mean squared error, on the other hand, exactly follows the expectation of Propositions 3.4 and 3.5 using μ as the uniform distribution on $\mathcal{X}^2$: in this case, $\mathbb{E}\|\tilde{f}\|_\mu \approx 0.66/D$, $\mathbb{E}\|\breve{f}\|_\mu \approx 0.83/D$. (This is natural, as the expectation is exact.) Convergence to that mean, however, is substantially faster than guaranteed by the McDiarmid bounds.

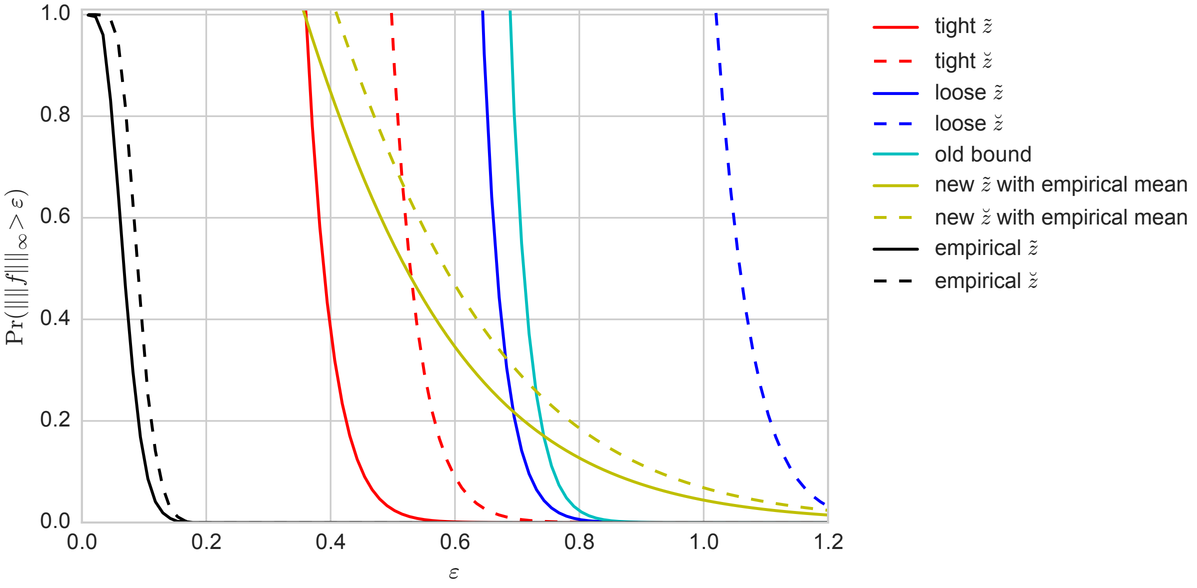


Figure 3.5: $\Pr(\mathbb{E}\|f\|_\infty > \varepsilon)$ for the Gaussian kernel on $[-3, 3]$ with $\sigma = 1$ and $D = 500$, based on 1 000 evaluations (black), numerical integration of the bounds from Propositions 3.6 and 3.7 (same colors as Figure 3.4), and the bounds of Propositions 3.10 and 3.11 using the empirical mean (yellow).

Chapter 4

Scalable distribution learning with approximate kernel embeddings

We now return to the distributional setting, developing embeddings for distributions in the style of — and employing — the random Fourier features studied in Chapter 3.

4.1 Mean map kernels

Armed with an approximate embedding for shift-invariant kernels on $\mathbb{R}^d$, we now need only a simple step to develop our first embedding for a distributional kernel, MMK. Recall that, given samples $\{X_i\}_{i=1}^n \sim P^n$ and $\{Y_j\}_{j=1}^m \sim Q^m$, $\textsc{mmk}(P, Q)$ can be estimated as

$$\textsc{mmk}(X, Y) = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} k(X_i, Y_j). \tag{4.1}$$

Simply plugging in an approximate embedding $z(x)^\mathsf{T} z(y) \approx k(x, y)$ yields

$$\textsc{mmk}(X, Y) \approx \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} z(X_i)^\mathsf{T} z(Y_j) = \left[\frac{1}{n} \sum_{i=1}^{n} z(X_i) \right]^\mathsf{T} \left[\frac{1}{m} \sum_{j=1}^{m} z(Y_j) \right] = \bar{z}(X)^\mathsf{T} \bar{z}(Y), \tag{4.2}$$

where we defined $\bar{z} : \mathcal{S} \to \mathbb{R}^D$ by $\bar{z}(X) := \frac{1}{n} \sum_{i=1}^{n} z(X_i)$. This additionally has a natural interpretation as the direct estimate of MMK in the Hilbert space induced by the feature map z, which approximates the Hilbert space associated with k.

Thus $\textsc{mmd}(P, Q) \approx \|\bar{z}(X) - \bar{z}(Y)\|$. Since this is simply a Euclidean distance, the generalized RBF kernel based on that distance $e^{-\gamma\, \textsc{mmd}^2}$ can be approximately embedded with $z(\bar{z}(\cdot))$.

This natural approximation has been considered many times quite recently (Mehta and Gray 2010; S. Li and Tsang 2011; Zhao and Meng 2014; Chwialkowski et al. 2015; Flaxman, Y.-X. Wang, et al. 2015; Jitkrittum, Gretton, et al. 2015; Lopez-Paz et al. 2015; Sutherland and Schneider 2015; Sutherland, J. B. Oliva, et al. 2016).

4.1.1 Convergence bounds

We will consider two approaches to proving bounds on this MMD embedding.

Applying uniform bounds

The following trivial bound allows the application of uniform convergence bounds to MMK estimators. Theorems 3 and 4 of Zhao and Meng (2014) appear to reduce to it.

Proposition 4.1 (Uniform convergence of $\bar{z}(X)^\mathsf{T}\bar{z}(Y)$)**.** *Let $z : \mathcal{X} \to \mathbb{R}^D$ be a random approximate embedding for a kernel k on some set $\mathcal{X}$ such that for some $\varepsilon > 0$, $0 < \delta \le 1$:*

$$\Pr\left(\sup_{x,y \in \mathcal{X}} \left| z(x)^\mathsf{T} z(y) - k(x,y) \right| \ge \varepsilon \right) \le \delta. \tag{4.3}$$

Define $\widehat{\textsc{mmk}} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ as the inner product between the mean maps under kernel k between the empirical distributions of the two inputs, as in (4.1). Let $\bar{z} : \mathcal{S} \to \mathbb{R}^D$ be given by $\bar{z}(X) := \frac{1}{n}\sum_{i=1}^n z(X_i)$. Then

$$\Pr\left(\sup_{X,Y \in \mathcal{S}} \left| \bar{z}(X)^\mathsf{T}\bar{z}(Y) - \widehat{\textsc{mmk}}(X,Y) \right| \ge \varepsilon \right) \le \delta.$$

Proof. For any $X, Y \subseteq \mathcal{X}$, we have

$$\begin{aligned} \left| \bar{z}(X)^\mathsf{T}\bar{z}(Y) - \textsc{mmk}(X,Y) \right| &= \left| \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m \left(z(X_i)^\mathsf{T} z(Y_j) - k(X_i, Y_j) \right) \right| \\ &\le \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m \left| z(X_i)^\mathsf{T} z(Y_j) - k(X_i, Y_j) \right|. \end{aligned}$$

If (4.3) holds, clearly this quantity is at most ε for all X, Y. □

Corollary 4.2 (Uniform convergence of $\|\bar{z}(X) - \bar{z}(Y)\|^2$ to the pairwise estimator)**.** *Let z, k, ε, δ, and $\widehat{\textsc{mmk}}$ be as in Proposition 4.1. Define $\widehat{\textsc{mmd}}_b : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ as $\widehat{\textsc{mmd}}_b(X,Y)^2 := \widehat{\textsc{mmk}}(X,X) + \widehat{\textsc{mmk}}(Y,Y) - 2\widehat{\textsc{mmk}}(X,Y)$. Then*

$$\Pr\left(\sup_{X,Y \in \mathcal{S}} \left| \|\bar{z}(X) - \bar{z}(Y)\|^2 - \widehat{\textsc{mmd}}_b^2(X,Y) \right| \ge 4\varepsilon \right) \le \delta.$$

Proof. With probability at least δ, each of $\left|\bar{z}(X)^\mathsf{T}\bar{z}(X) - \widehat{\textsc{mmk}}(X,X)\right|$, $\left|\bar{z}(Y)^\mathsf{T}\bar{z}(Y) - \widehat{\textsc{mmk}}(Y,Y)\right|$, and $\left|\bar{z}(X)^\mathsf{T}\bar{z}(Y) - \widehat{\textsc{mmk}}(X,Y)\right|$ are at most ε by Proposition 4.1, □

Proposition 4.3 (Convergence of $\|\bar{z}(X) - \bar{z}(Y)\|$ to the MMD)**.** *Let z, k, $\varepsilon_{\bar{z}}$, δ, and $\widehat{\textsc{mmk}}$ be as in Proposition 4.1, but additionally requiring that $k(x,y) \ge 0$ for all $x, y \in \mathcal{X}$. Fix a pair of input distributions P, Q over $\mathcal{X}$. Take $X \sim P^n$, $Y \sim Q^m$; then for any $\varepsilon_{\textsc{mmd}} > 0$ we have*

$$\Pr_{X,Y,\bar{z}}\left(\left| \|\bar{z}(X) - \bar{z}(Y)\| - \textsc{mmd}(P,Q) \right| > \frac{2}{\sqrt{m}} + \frac{2}{\sqrt{n}} + \varepsilon_{\textsc{mmd}} + 32\varepsilon_{\bar{z}} \right) \le 2\exp\left(-\frac{m\,n\,\varepsilon_{\textsc{mmd}}^2}{2(m+n)} \right) + \delta.$$

Proof. For the sake of brevity, let $\eta_{\bar{z}} := \|\bar{z}(X) - \bar{z}(Y)\|$, $\eta_{XY} := \widehat{\textsc{mmd}}_b(X, Y)$, $\eta_{PQ} := \textsc{mmd}(P, Q)$, $c_{mn} := \frac{2}{\sqrt{m}} + \frac{2}{\sqrt{n}}$. Thus we wish to bound

$$\begin{aligned}\Pr_{X,Y,\bar{z}}\left(\left|\eta_{\bar{z}} - \eta_{PQ}\right| > c_{mn} + \varepsilon_{\textsc{mmd}} + 32\varepsilon_{\bar{z}}\right) &\le \Pr_{X,Y,\bar{z}}\left(\left|\eta_{\bar{z}} - \eta_{XY}\right| + \left|\eta_{XY} - \eta_{PQ}\right| > c_{mn} + \varepsilon_{\textsc{mmd}} + 32\varepsilon_{\bar{z}}\right) \\ &\le \Pr_{X,Y,\bar{z}}\left(\left|\eta_{\bar{z}} - \eta_{XY}\right| > 32\varepsilon_{\bar{z}}\right) + \Pr_{X,Y}\left(\left|\eta_{XY} - \eta_{PQ}\right| > c_{mn} + \varepsilon_{\textsc{mmd}}\right).\end{aligned}$$

Theorem 7 of Gretton, Borgwardt, et al. (2012) bounds the latter term:

$$\Pr_{X,Y}\left(\left|\eta_{XY} - \eta_{PQ}\right| > c_{mn} + \varepsilon_{\textsc{mmd}}\right) \le 2\exp\left(-\frac{m\,n\,\varepsilon_{\textsc{mmd}}^2}{2(m+n)}\right).$$

For the former, note that $\eta_{\bar{z}}^2$ and η_{XY}^2 are each in $[0, 4]$, so

$$\left|\eta_{\bar{z}}^2 - \eta_{XY}^2\right| = \left|\eta_{\bar{z}} - \eta_{XY}\right| \left|\eta_{\bar{z}} + \eta_{XY}\right| \le 8\left|\eta_{\bar{z}} - \eta_{XY}\right|.$$

Thus by Corollary 4.2,

$$\Pr_{X,Y,\bar{z}}\left(\left|\eta_{\bar{z}} - \eta_{XY}\right| > 32\varepsilon_{\bar{z}}\right) \le \Pr_{X,Y,\bar{z}}\left(\left|\eta_{\bar{z}}^2 - \eta_{XY}^2\right| > 4\varepsilon_{\bar{z}}\right) = \mathbb{E}_{X,Y}\Pr_{\bar{z}}\left(\left|\eta_{\bar{z}}^2 - \eta_{XY}^2\right| > 4\varepsilon_{\bar{z}}\right) \le E_{X,Y}\delta = \delta. \quad\square$$

Proposition 4.4 (Convergence of kernel approximation for a given P, Q)**.** *Let z, $\bar{z}$, k, $\varepsilon_{\bar{z}}$, δ, $\widehat{\textsc{mmk}}$, P, Q, X, Y, n, and m be as in Proposition 4.3, with the z embedding into dimension D_1. Define a kernel on distributions $K(P, Q) := \exp\left(-\frac{1}{2\sigma^2}\textsc{mmd}^2(P, Q)\right)$ for some bandwidth $\sigma > 0$. Let $k_\sigma(x, y) := \exp\left(-\frac{1}{2\sigma^2}\|x - y\|^2\right)$ be the Gaussian* RBF *kernel of bandwidth σ, and z_σ its embedding using either $\tilde{z}$ or $\breve{z}$ with embedding dimension D_2. Estimate the kernel $K(P, Q)$ as $z_\sigma(\bar{z}(X))^\mathsf{T} z_\sigma(\bar{z}(Y))$. Then for any $\varepsilon_{\textsc{mmd}} > 0$, ε_{z_σ}:*

$$\begin{aligned}\Pr_{X,Y,\bar{z},z_\sigma}\left(\left|z_\sigma(\bar{z}(X))^\mathsf{T} z_\sigma(\bar{z}(Y)) - K(P, Q)\right| > \frac{1}{\sigma\sqrt{e}}\left(\frac{2}{\sqrt{m}} + \frac{2}{\sqrt{n}} + \varepsilon_{\textsc{mmd}} + 32\varepsilon_{\bar{z}}\right) + \varepsilon_{z_\sigma}\right)& \\ \le 2\exp\left(-\frac{m\,n\,\varepsilon_{\textsc{mmd}}^2}{2(m+n)}\right) + \delta + 2\exp\left(-\frac{D_2\varepsilon_{z_\sigma}^2}{8 + \frac{8}{3}\varepsilon_{z_\sigma}}\right)&.\end{aligned}$$

Proof. Define $r_\sigma : \mathbb{R} \to \mathbb{R}$ by $r_\sigma(x) := \exp\left(-x^2/(2\sigma^2)\right)$. Let $\eta_{\bar{z}} := \|\bar{z}(X) - \bar{z}(Y)\|$, $\eta_{PQ} := \textsc{mmd}(P, Q)$. Then the error in question is

$$\left|r_\sigma(\eta_{PQ}) - z_\sigma(\bar{z}(X))^\mathsf{T} z_\sigma(\bar{z}(Y))\right| \le \left|r_\sigma(\eta_{PQ}) - r_\sigma(\eta_{\bar{z}})\right| + \left|r_\sigma(\eta_{\bar{z}}) - z_\sigma(\bar{z}(X))^\mathsf{T} z_\sigma(\bar{z}(Y))\right|.$$

The first term, because r_σ is $\frac{1}{\sigma\sqrt{e}}$-Lipschitz, is at most $\frac{1}{\sigma\sqrt{e}}\left|\eta_{PQ} - \eta_{\bar{z}}\right|$. Using Proposition 4.3:

$$\Pr_{X,Y,\bar{z}}\left(\left|\eta_{PQ} - \eta_{\bar{z}}\right| > \frac{2}{\sqrt{m}} + \frac{2}{\sqrt{n}} + \varepsilon_{\textsc{mmd}} + 32\varepsilon_{\bar{z}}\right) \le 2\exp\left(-\frac{m\,n\,\varepsilon_{\textsc{mmd}}^2}{2(m+n)}\right) + \delta.$$

The latter term is just the error of the z_σ embedding on the inputs $\bar{z}(X)$, $\bar{z}(Y)$. We can use the Bernstein bound of (B.3) and (B.6), simplifying it a bit because $\operatorname{Var}[\cos(\omega^\mathsf{T}\Delta)] \le \frac{1}{2}$ for pixie kernels:

$$\begin{aligned}
&\Pr_{X,Y,\bar{z},z_\sigma}\left(\left|r_\sigma(\eta_{\bar{z}}) - z_\sigma(\bar{z}(X))^\mathsf{T} z_\sigma(\bar{z}(Y))\right| > \varepsilon_{z_\sigma}\right) \\
&\qquad = \mathbb{E}_{X,Y,\bar{z}} \Pr_{z_\sigma}\left(\left|r_\sigma(\eta_{\bar{z}}) - z_\sigma(\bar{z}(X))^\mathsf{T} z_\sigma(\bar{z}(Y))\right| > \varepsilon_{z_\sigma}\right) \\
&\qquad \le \mathbb{E}_{X,Y,\bar{z}}\, 2\exp\left(-\frac{D_2\varepsilon_{z_\sigma}^2}{8+\frac{8}{3}\varepsilon_{z_\sigma}}\right) = 2\exp\left(-\frac{D_2\varepsilon_{z_\sigma}^2}{8+\frac{8}{3}\varepsilon_{z_\sigma}}\right). \qquad \square
\end{aligned}$$

It is worth re-emphasizing two points: first, that the δ of these bounds will depend on the diameter of $\mathcal{X}$, and so they are not directly applicable to distributions on unbounded domains. Secondly, extension of Proposition 4.4 to a bound uniform over input distributions would require a uniform version of Proposition 4.3, presumably based on a uniform extension of Theorem 7 of Gretton, Borgwardt, et al. (2012). This could be done e.g. by bounding the Lipschitz constant of the error of the mmd estimator over some smoothness class of distributions, as in the proof of Proposition 3.6.

For fixed inputs

We can also show bounds more directly for a fixed pair of inputs (fixed sample sets X, Y at first; later, for fixed distributions P, Q). This approach will allow us to consider unbounded domains, but does not allow for direct uniform results as in Proposition 4.1 and Corollary 4.2.

Proposition 4.5 (Convergence of $\bar{z}(X)^\mathsf{T}\bar{z}(Y)$ for fixed X, Y)**.** *Let $z : \mathcal{X} \to \mathbb{R}^D$ be either $\tilde{z}$ of (3.1) or $\breve{z}$ of (3.2), corresponding to a continuous, shift-invariant, positive definite kernel function $k(x, y) = \underline{k}(x - y)$ with $\underline{k}(0) = 1$. Let $\bar{z}(X) := \frac{1}{n}\sum_{i=1}^n z(X_i)$. Then, considering $X \subseteq \mathcal{X}$ of size n and $Y \subseteq \mathcal{X}$ of size m fixed:*

(i) The variance of the mmk embedding is:

$$\operatorname{Var}\left[\bar{z}(X)^\mathsf{T}\bar{z}(Y)\right] = \frac{1}{n^2m^2}\sum_{i,j}\sum_{i',j'}\operatorname{Cov}(z(X_i)^\mathsf{T}z(Y_j), z(X_{i'})^\mathsf{T}z(Y_{j'})),$$

which for $\tilde{z}$ is

$$\begin{aligned}
\tilde{V}_{X,Y} := \frac{1}{D}\tilde{v}_{X,Y} := \frac{1}{D}\frac{1}{n^2m^2}\sum_{i,j}\sum_{i',j'}\big[&\underline{k}(X_i - X_{i'} - Y_j + Y_{j'}) + \underline{k}(X_i + X_{i'} - Y_j - Y_{j'}) \\
&-2\underline{k}(X_i - Y_j)\underline{k}(X_{i'} - Y_{j'})\big]
\end{aligned} \tag{4.4}$$

and for $\breve{z}$ is

$$\begin{aligned}
\breve{V}_{X,Y} := \frac{1}{D}\breve{v}_{X,Y} := \frac{1}{D}\frac{1}{n^2m^2}\sum_{i,j}\sum_{i',j'}\big[&\tfrac{1}{2}\underline{k}(X_i - X_{i'} - Y_j + Y_{j'}) + \tfrac{1}{2}\underline{k}(X_i + X_{i'} - Y_j - Y_{j'}) \\
&-\underline{k}(X_i - Y_j)\underline{k}(X_{i'} - Y_{j'}) + \tfrac{1}{2}\underline{k}(X_i - X_{i'} + Y_j - Y_{j'})\big].
\end{aligned} \tag{4.5}$$

Note that

$$\breve{\nu}_{X,Y} = \frac{1}{2}\left(\tilde{\nu}_{X,Y} + \frac{1}{n^2m^2}\sum_{i,j}\sum_{i',j'}\underline{k}(X_i - X_{i'} + Y_j - Y_{j'})\right).$$

(ii) Let $\tilde{\alpha}^{(\varepsilon)}_{X,Y} := \min\left(4, 2\tilde{\nu}_{X,Y} + \frac{4}{3}\varepsilon\right)$*, using the variance factor* $\tilde{\nu}_{X,Y}$ *of (4.4). Similarly use (4.5) to define* $\breve{\alpha}^{(\varepsilon)}_{X,Y} := \min\left(8, 2\breve{\nu}_{X,Y} + \frac{4}{3}\varepsilon\right)$*. Then, letting* $\alpha^{(\varepsilon)}_{X,Y}$ *denote* $\tilde{\alpha}^{(\varepsilon)}_{X,Y}$ *for the* $\tilde{z}$ *embedding and* $\breve{\alpha}^{(\varepsilon)}_{X,Y}$ *for the* $\breve{z}$ *embedding:*

$$\Pr\left(\left|\bar{z}(X)^{\mathsf{T}}\bar{z}(Y) - \widehat{\textsc{mmk}}(X,Y)\right| \geq \varepsilon\right) \leq 2\exp\left(-\frac{D\varepsilon^2}{\alpha^{(\varepsilon)}_{X,Y}}\right).$$

(iii) Let $\alpha^{(\infty)}$ *be* 4 *for the* $\tilde{z}$ *embedding and* 8 *for* $\breve{z}$*. Then*

$$\mathbb{E}\left|\bar{z}(X)^{\mathsf{T}}\bar{z}(Y) - \widehat{\textsc{mmk}}(X,Y)\right| \leq \sqrt{\frac{\alpha^{(\infty)}\pi}{D}}.$$

Proof.

(i) Simply expand $\bar{z}(X)^{\mathsf{T}}\bar{z}(Y)$ into a sum, as in (4.2), and use (3.3) and (3.5).

(ii) For $\tilde{z}$, we can think of $\bar{z}(X)^{\mathsf{T}}\bar{z}(Y)$ as an average of $\frac{D}{2}$ terms like

$$\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\cos\left(\omega^{\mathsf{T}}(X_i - Y_j)\right),$$

each of which has mean $\widehat{\textsc{mmk}}(X,Y)$, variance $\frac{1}{2}\tilde{\nu}_{X,Y}$, and is bounded by $[-1,1]$. The claim gives the better of Hoeffding's and Bernstein's inequalities; the latter is tighter when $\varepsilon < 3 - \frac{3}{2}\tilde{\nu}_{X,Y}$.
Similarly, $\breve{z}$ gives an average of D terms like

$$\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\left[\cos\left(\omega^{\mathsf{T}}(X_i - Y_j)\right) + \cos\left(\omega^{\mathsf{T}}(X_i + Y_j) + 2b\right)\right],$$

each of which has mean $\widehat{\textsc{mmk}}(X,Y)$, variance $\breve{\nu}_{X,Y}$, and is bounded by $[-2,2]$. Here Bernstein's is tighter for $\varepsilon < 6 - \frac{3}{2}\breve{\nu}_{X,Y}$.

(iii) Integrate the Hoeffding-form bound of (ii), using $\mathbb{E}|X| = \int_0^\infty \Pr(|X| \geq \varepsilon)\,\mathrm{d}\varepsilon$. □

Note that Proposition 4.5(i) gives that the variance in terms of D is exactly $\frac{\nu_{X,Y}}{D}$ (with $\nu_{X,Y}$ depending only on k, X, and Y), whereas Proposition 4.1 does not allow for an easy form for the variance when used with Propositions 3.6 and 3.7.

We can easily extend this to a convergence bound on the MMD embedding. The variance is also available via the same technique as Proposition 4.5(i), and is still $O(1/D)$.

Corollary 4.6 (Convergence of $\|\bar{z}(X) - \bar{z}(Y)\|^2$ for fixed X, Y)**.** *Let z, $\bar{z}$, k, X, Y, m, n, and $\alpha^{(\infty)}$ be as in Proposition 4.5. Define $\widehat{\textsc{mmd}}_b$ as in Corollary 4.2. Then:*

$$\Pr\left(\left| \|\bar{z}(X) - \bar{z}(Y)\|^2 - \widehat{\textsc{mmd}}_b^2(X, Y) \right| > \varepsilon \right) \le 6 \exp\left(-\frac{D\varepsilon^2}{16\alpha^{(\infty)}} \right).$$

Proof. We can upper-bound $\left| \|\bar{z}(X) - \bar{z}(Y)\|^2 - \widehat{\textsc{mmd}}_b(X, Y)^2 \right|$ by

$$\left|\bar{z}(X)^\mathsf{T} \bar{z}(X) - \widehat{\textsc{mmk}}(X, X)\right| + \left|\bar{z}(Y)^\mathsf{T} \bar{z}(Y) - \widehat{\textsc{mmk}}(Y, Y)\right| + 2\left|\bar{z}(X)^\mathsf{T} \bar{z}(Y) - \widehat{\textsc{mmk}}(X, Y)\right|.$$

Use the Hoeffding version of Proposition 4.5(ii) with $\frac{1}{4}\varepsilon$ for each term, then a union bound. □

We can also allow $X \sim P$, $Y \sim Q$ to be random:

Corollary 4.7 (Variance of $\bar{z}(X)^\mathsf{T} \bar{z}(Y)$ for random X, Y)**.** *Let z, $\bar{z}$, k be as in Proposition 4.5. Let $\textsc{mmk}$ denote the inner product between mean embeddings with the kernel k. Fix distributions P, Q over $\mathcal{X}$. Letting $X, X' \overset{iid}{\sim} P$, denote the distribution of $X - X'$ as Δ_P and $X + X'$ as T_P. Similarly define Δ_Q and T_Q. Then the expected variance of the embedding-based $\textsc{mmk}$ estimator for $\tilde{z}$ is:*

$$\tilde{V}_{P,Q} := \mathbb{E}_{X\sim P, Y\sim Q} \operatorname{Var}\left[\bar{z}(X)^\mathsf{T} \bar{z}(Y)\right] = \frac{1}{D}\left[\textsc{mmk}\left(\Delta_P, \Delta_Q\right) + \textsc{mmk}\left(T_P, T_Q\right) - 2\,\textsc{mmk}\left(P, Q\right)^2\right],$$

and for $\breve{z}$ is:

$$\breve{V}_{P,Q} := \mathbb{E}_{X\sim P, Y\sim Q} \operatorname{Var}\left[\bar{z}(X)^\mathsf{T} \bar{z}(Y)\right] = \frac{1}{D}\left[\textsc{mmk}\left(\Delta_P, \Delta_Q\right) + \tfrac{1}{2}\,\textsc{mmk}\left(T_P, T_Q\right) - \textsc{mmk}\left(P, Q\right)^2\right].$$

Note that $V_{P,Q}$ is not the "full" variance of the estimator, which is

$$\operatorname{Var}_{X,Y,\bar{z}}\left[\bar{z}(X)^\mathsf{T} \bar{z}(Y)\right] = V_{P,Q} + \operatorname{Var}_{X,Y} \widehat{\textsc{mmk}}(X, Y).$$

Proof. For the values of $V_{P,Q}$, take expectations of Proposition 4.5(i). The final statement is just the law of total variance, noting that $\mathbb{E}_{\bar{z}}\left[\bar{z}(X)^\mathsf{T} \bar{z}(Y)\right] = \widehat{\textsc{mmk}}(X, Y)$. □

Corollary 4.8 (Convergence of $\|\bar{z}(X) - \bar{z}(Y)\|$ for random X, Y)**.** *Let z, $\bar{z}$, k be as in Proposition 4.5, but additionally require that $k(x, y) \ge 0$ for all x, y. Let $\textsc{mmd}$ denote the maximum mean discrepancy with kernel k. Fix distributions P, Q over $\mathcal{X}$, and let $X \sim P^n$ and $Y \sim Q^m$. Let $\alpha^{(\infty)}$ be 4 for $\tilde{z}$ and 8 for $\breve{z}$. Then for any $\varepsilon_{\textsc{mmd}}, \varepsilon_{\bar{z}} > 0$,*

$$\Pr_{X,Y,\bar{z}}\left(\left| \|\bar{z}(X) - \bar{z}(Y)\| - \textsc{mmd}(P, Q) \right| > \frac{2}{\sqrt{m}} + \frac{2}{\sqrt{n}} + \varepsilon_{\textsc{mmd}} + \varepsilon_{\bar{z}} \right)$$
$$\le 2\exp\left(-\frac{\varepsilon_{\textsc{mmd}}^2 mn}{8(m+n)} \right) + 6\exp\left(-\frac{D\varepsilon_{\bar{z}}^2}{1024\alpha^{(\infty)}} \right).$$

Proof. The argument is as for Proposition 4.3, replacing Corollary 4.2 with Corollary 4.6. □

Corollary 4.9 (Convergence of kernel approximation for a given P, Q)**.** *Let z, $\bar{z}$, k, X, Y, and $\alpha^{(\infty)}$ be as in Corollary 4.8, with z having embedding dimension D_1. Define a kernel $K(P, Q) := \exp\left(-\frac{1}{2\sigma^2} \operatorname{MMD}^2(P, Q)\right)$ for some bandwidth $\sigma > 0$. Let z_σ be the embedding for the Gaussian RBF kernel of bandwidth σ, using either $\tilde{z}$ or $\breve{z}$ of embedding dimension D_2; define the estimator of $K(P, Q)$ as $\hat{K}(X, Y) := z_\sigma(\bar{z}(X))^\mathsf{T} z_\sigma(\bar{z}(Y))$. Then for any $\varepsilon_{\mathrm{MMD}}, \varepsilon_{\bar{z}}, \varepsilon_{z_\sigma} > 0$:*

$$\Pr_{X,Y,\bar{z}}\left(\left|z_\sigma(\bar{z}(X))^\mathsf{T} z_\sigma(\bar{z}(Y)) - K(P, Q)\right| > \frac{1}{\sigma\sqrt{e}}\left(\frac{2}{\sqrt{m}} + \frac{2}{\sqrt{n}} + \varepsilon_{\mathrm{MMD}} + \varepsilon_{\bar{z}}\right) + \varepsilon_{RBF}\right)$$
$$\leq 2\exp\left(-\frac{mn\varepsilon_{\mathrm{MMD}}^2}{32(m+n)}\right) + 6\exp\left(-\frac{D_1\varepsilon_{\bar{z}}^2}{1024\alpha^{(\infty)}}\right) + 2\exp\left(-\frac{D_2\varepsilon_{z_\sigma}^2}{8 + \frac{8}{3}\varepsilon_{z_\sigma}}\right).$$

Proof. As for Proposition 4.4, using Corollary 4.8 rather than Proposition 4.3. □

Converting Corollary 4.9 to a bound uniform over distributions would have similar challenges to those of Proposition 4.4, except that the $\varepsilon_{\bar{z}}$ term would similarly need to be treated over a smoothness class of distributions, whereas Proposition 4.4 gets that "for free" via Corollary 4.2.

4.2 L_2 distances

J. B. Oliva, Neiswanger, et al. (2014) gave an embedding for $e^{-\gamma L_2^2}$, by first embedding L_2 with orthonormal projections and then applying random Fourier features.

Suppose that $\mathcal{X} \subseteq [0, 1]^d$. Let $\{\varphi_\alpha\}_{\alpha \in \mathbb{Z}^d}$ be an orthonormal basis for $L_2([0, 1]^d)$, perhaps constructed as the d-fold tensor product of an orthonormal basis for $L_2([0, 1])$. Then any function $f \in L_2([0, 1]^d)$ can be represented as $f(x) = \sum_{\alpha \in \mathbb{Z}^d} a_\alpha(f)\varphi_\alpha(x)$, where

$$a_\alpha(f) := \langle \varphi_\alpha, f \rangle = \int_{[0,1]^d} \varphi_\alpha(t) f(t) \,\mathrm{d}t,$$

and for any $f, g \in L_2([0, 1]^d)$,

$$\begin{aligned}\langle f, g \rangle &= \left\langle \sum_{\alpha \in \mathbb{Z}^d} a_\alpha(f)\varphi_\alpha, \sum_{\beta \in \mathbb{Z}^d} a_\beta(g)\varphi_\beta \right\rangle \\ &= \sum_{\alpha \in \mathbb{Z}^d} \sum_{\beta \in \mathbb{Z}^d} a_\alpha(f) a_\beta(g) \langle \varphi_\alpha, \varphi_\beta \rangle \\ &= \sum_{\alpha \in \mathbb{Z}^d} a_\alpha(f) a_\alpha(g).\end{aligned}$$

Let $V \subset \mathbb{Z}^d$ be an appropriately chosen finite set of indices $\{\alpha_1, \ldots, \alpha_{|V|}\}$. Define $\vec{a}(f) = (a_{\alpha_1}(f), \ldots, a_{\alpha_{|V|}}(f))^\mathsf{T} \in \mathbb{R}^{|V|}$. If f and g are smooth with respect to V, i.e. they have only small contributions from basis functions not in V, we have

$$\langle f, g \rangle = \sum_{\alpha \in \mathbb{Z}^d} a_\alpha(f)\, a_\alpha(g) \approx \sum_{\alpha \in V} a_\alpha(f)\, a_\alpha(g) = \vec{a}(f)^\mathsf{T}\, \vec{a}(g).$$

Now, given a sample $X = \{X_1, \dots, X_n\} \sim P^n$, let $\hat{P}(x) = \frac{1}{n} \sum_{i=1}^n \delta(X_i - x)$ be the empirical distribution of X. J. B. Oliva, Neiswanger, et al. (2014) estimate the density p as

$$\hat{p}(x) = \sum_{\alpha \in V} a_\alpha(\hat{P})\, \varphi_\alpha(x) \qquad \text{where } a_\alpha(\hat{P}) = \int_{[0,1]^d} \varphi_\alpha(t)\, \mathrm{d}\hat{P}(t) = \frac{1}{n} \sum_{i=1}^n \varphi_\alpha(X_i). \tag{4.6}$$

Note that technically this is an extension of a_α to a broader domain than $L_2([0,1]^d)$. Assuming that the distribution functions are smooth with respect to V, i.e. they lie in the Sobolev ellipsoid corresponding to the basis functions of V, we thus have that

$$\langle p, q \rangle \approx \langle \hat{p}, \hat{q} \rangle \approx \vec{a}(\hat{P})^\mathsf{T}\, \vec{a}(\hat{Q})$$

and so

$$z(\vec{a}(\hat{P}))^\mathsf{T} z(\vec{a}(\hat{Q})) \approx \exp\left(-\frac{1}{2\sigma^2} \|P - Q\|_2^2 \right).$$

For the Sobolev assumption to hold on a fairly general class of distributions, however, we need $|V|$ to be $\Omega(T^d)$ for some constant T. Since the embedding is of dimension $|V|$, this method is limited in practice to fairly low dimensions d.

J. B. Oliva, Neiswanger, et al. (2014) proved learning theoretic bounds on the use of this estimator with ridge regression. Because the L_2 embedding is deterministic, the convergence portion of the bound is not especially interesting: the Sobolev assumption on the densities is essentially that the embedding error is bounded by a certain amount.

4.2.1 Connection to MMD embedding

The components of the embedding (4.6) are of the form

$$a_\alpha(X) = \frac{1}{n} \sum_{i=1}^n \varphi_\alpha(X_i),$$

whereas the embedding $\bar{z}$ of Section 4.1 has components of the form

$$\bar{z}(X)_j = \frac{1}{n} \sum_{i=1}^n z_j(X_i).$$

This similarity in form is tantalizing, but how similar are the z_j and φ_α functions?

Taking a more general view of the MMD embedding than solely one based on random Fourier features, the L_2 embedding can be viewed as proportional to a mean map embedding in the Hilbert space defined by the basis functions $\{\varphi_\alpha\}_{\alpha \in V}$, with a kernel given by $k(x, y) = \sum_{\alpha \in V} \varphi_\alpha(x) \varphi_\alpha(y)$. As V expands to $\mathbb{Z}^d$, this space converges to $L_2([0,1]^d)$, with a shift-invariant kernel of the Dirac delta function.

In practice, we often use the tensor product of the cosine, Fourier, or trigonometric bases for $L_2([0,1])$. However, the following orthonormal basis[1] for $L_2([0,1]^d)$ more closely resembles a mean map embedding with the $\tilde{z}$ random Fourier features:

$$\varphi_0(x) = 1 \qquad \varphi_k(x) = \sqrt{2} \cos(2\pi k^\mathsf{T} x), k \in \mathcal{K}_d \qquad \varphi'_k(x) = \sqrt{2} \sin(2\pi k^\mathsf{T} x), k \in \mathcal{K}_d$$

[1] This is not in standard use, but we can see that its span is dense in L_2 via the Stone-Weierstrass theorem.

where $\mathcal{K}_d$ is the set of d-vectors with integral entries with at least one nonzero coordinate, the first of which is positive: $\mathcal{K}_1 = \{1, 2, \dots\}$, $\mathcal{K}_d = (\{1, 2, \dots\} \times \mathbb{Z}^{d-1}) \cup (\{0\} \times \mathcal{K}_{d-1})$. This restriction is needed for orthogonality because $\varphi_k = \varphi_{-k}$, and $\varphi'_k = -\varphi'_{-k}$. We can obtain an almost exactly equivalent L_2 embedding, however, by using $\mathcal{K}'_d = \mathbb{Z}^d \setminus 0$: inner products are then effectively doubled, except for the constant term 1. Consider the index set $V_T = \{0\} \cup \{k \in \mathcal{K}'_d : \max_j |k_j| \le T\}$. Now, note that for kernels whose Fourier transforms are discrete distributions, sampling without replacement in the $\tilde{z}$ embedding still works: tighter versions of many of the same bounds even hold, replacing the Hoeffding or Bernstein bounds with their Serfling-style analogues (Serfling 1974; Bardenet and Maillard 2015). Thus the $\tilde{z}$ embedding for a kernel corresponding to the Fourier transform of a uniform distribution over $[-T, T]^d$ has the exact same arguments to the sine and cosine terms, except for adding a useless constant 0 dimension. This $\tilde{z}$ embedding is of dimension $D = 2(2T+1)^d$, and is scaled by $\frac{1}{\sqrt{D}}$ relative to the L_2 embedding. The kernel being embedded is the tensor product of a normalized Dirichlet kernel on each dimension, namely

$$\underline{k}(\Delta) = \prod_{j=1}^{d} \frac{1}{2T+1} \left(1 + 2 \sum_{k=1}^{T} \cos(2\pi k \Delta_j) \right) = \prod_{j=1}^{d} \frac{\sin\left((2T+1)\,\pi\Delta_j\right)}{(2T+1)\sin(\pi\Delta_j)}.$$

The Dirichlet kernel is well-known in the theory of Fourier transforms, and is an approximation to the Dirac δ function. Note also that Corollary 4(ii) of Sriperumbudur, Gretton, et al. (2010) shows that as $T \to \infty$, the MMD based on $\underline{k}$ converges to the appropriate rescaling constant times the L_2 distance, independently confirming that the L_2 embedding asymptotically works.

4.3 Information-theoretic distances

We will now show how to extend this general approach to a class of information theoretic distances that includes TV, JS, and squared Hellinger (Sutherland, J. B. Oliva, et al. 2016). We consider a class of metrics that we term *homogeneous density distances* (HDDS):

$$\rho^2(p, q) = \int_{[0,1]^d} \kappa(p(x), q(x))\,\mathrm{d}x \tag{4.7}$$

where $\kappa : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ is a 1-homogenous negative-definite function[2]. That is, $\kappa(tx, ty) = t\kappa(x, y)$ for all $t > 0$, and there exists some Hilbert space where $\|x - y\|^2 = \kappa(x, y)$. This class was studied by Fuglede (2005); Table 4.1 shows some important instances.

Our embedding will take three steps:

Embedding HDDS into L_2 We define a random function ψ such that $\rho(p, q) \approx \|\psi(p) - \psi(q)\|$, where $\psi(p)$ is a function from $[0, 1]^d$ to $\mathbb{R}^{2M}$. Thus the metric space of densities with distance ρ is approximately embedded into the metric space of $2M$-dimensional L_2 functions.

Finite Embeddings of L_2 We use the approach of Section 4.2 to approximately embed smooth L_2 functions into finite vectors in $\mathbb{R}^{|V|}$. Combined with the previous step, we obtain features $A(p) \in \mathbb{R}^{2M|V|}$ such that ρ is approximated by Euclidean distances between the $A(\cdot)$ features.

[2] Sometimes referred to as a negative-definite kernel.

Name	$\kappa(p(x), q(x))$	$\mathrm{d}\mu(\lambda)$
Jensen-Shannon (JS)	$\frac{p(x)}{2} \log\left(\frac{2p(x)}{p(x)+q(x)}\right) + \frac{q(x)}{2} \log\left(\frac{2q(x)}{p(x)+q(x)}\right)$	$\frac{\mathrm{d}\lambda}{\cosh(\pi\lambda)(1+\lambda^2)}$
Squared Hellinger (H^2)	$\frac{1}{2}\left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2$	$\frac{1}{2}\delta(\lambda = 1)\mathrm{d}\lambda$
Total Variation (TV)	$\lvert p(x) - q(x) \rvert$	$\frac{2}{\pi} \frac{\mathrm{d}\lambda}{1+4\lambda^2}$

Table 4.1: Various squared HDDS; $\mathrm{d}\mu$ will be defined shortly.

Embedding RBF Kernels into $\mathbb{R}^D$ We use random Fourier features $z(\cdot)$ so that inner products between $z(A(\cdot))$ features, in $\mathbb{R}^D$, approximate $K(p, q)$.

Vedaldi and Zisserman (2012) studied embeddings of a similar class of kernels, but only for discrete distributions (e.g. histograms). Their approach was basically analogous to ours, but uses a fixed sampling scheme rather than the random one we employ to approximate κ, and the L_2 embedding step is trivial in their setting since they operate componentwise. We compare to their approaches in Section 5.2, a case in which the histogram assumption harms the convergence of the estimator significantly with low sample sizes, but allows for faster computation.

Our embedding proceeds as follows. Fuglede (2005) shows that κ corresponds to a unique bounded measure $\mu(\lambda)$, shown in Table 4.1, by

$$\kappa(x, y) = \int_{\mathbb{R}_{\geq 0}} \lvert x^{\frac{1}{2}+\mathbb{i}\lambda} - y^{\frac{1}{2}+\mathbb{i}\lambda} \rvert^2 \, \mathrm{d}\mu(\lambda).$$

The following is equivalent, but makes it easier to find μ:

$$\kappa(x, 1/x) = Zx + Z\frac{1}{x} - 2 \int_{\mathbb{R}_{\geq 0}} \cos(2\lambda \log x) \, \mathrm{d}\mu(\lambda). \tag{4.8}$$

Let $Z := \mu(\mathbb{R}_{\geq 0})$ so that μ/Z is a distribution; also define $c_\lambda := (-\frac{1}{2} + \mathbb{i}\lambda)/(\frac{1}{2} + \mathbb{i}\lambda)$. Then

$$\kappa(x, y) = \mathbb{E}_{\lambda \sim \frac{\mu}{Z}} \lvert g_\lambda(x) - g_\lambda(y) \rvert^2 \qquad \text{where } g_\lambda(x) := \sqrt{Z} c_\lambda \left(x^{\frac{1}{2}+\mathbb{i}\lambda} - 1 \right).$$

We approximate the expectation with an empirical mean. Let $\lambda_j \overset{iid}{\sim} \frac{\mu}{Z}$ for $j \in \{1, \ldots, M\}$; then

$$\kappa(x, y) \approx \frac{1}{M} \sum_{j=1}^{M} \lvert g_{\lambda_j}(x) - g_{\lambda_j}(y) \rvert^2.$$

Hence, the squared HDD is, letting $\Re, \Im$ denote the real and imaginary parts:

$$\begin{aligned} \rho^2(p, q) &= \int_{[0,1]^d} \kappa(p(x), q(x)) \, \mathrm{d}x \\ &= \int_{[0,1]^d} \mathbb{E}_{\lambda \sim \frac{\mu}{Z}} \lvert g_\lambda(p(x)) - g_\lambda(q(x)) \rvert^2 \, \mathrm{d}x \\ &\approx \frac{1}{M} \sum_{j=1}^{M} \int_{[0,1]^d} \left(\left(\Re(g_{\lambda_j}(p(x))) - \Re(g_{\lambda_j}(q(x))) \right)^2 + \left(\Im(g_{\lambda_j}(p(x))) - \Im(g_{\lambda_j}(q(x))) \right)^2 \right) \mathrm{d}x \end{aligned}$$

$$= \frac{1}{M} \sum_{j=1}^{M} \|p_{\lambda_j}^R - q_{\lambda_j}^R\|^2 + \|p_{\lambda_j}^I - q_{\lambda_j}^I\|^2, \tag{4.9}$$

where

$$p_\lambda^R(x) := \Re(g_\lambda(p(x))), \qquad p_\lambda^I(x) := \Im(g_\lambda(p(x))).$$

Each p_λ function is in $L_2([0,1]^d)$, so we can approximate the Gaussian RBF kernel based on ρ, $\exp(-\gamma\rho^2(p,q))$, as in Section 4.2: let

$$A(P) := \frac{1}{\sqrt{M}} \left(\vec{a}(p_{\lambda_1}^R)^\mathsf{T}, \vec{a}(p_{\lambda_1}^I)^\mathsf{T}, \dots, \vec{a}(p_{\lambda_M}^R)^\mathsf{T}, \vec{a}(p_{\lambda_M}^I)^\mathsf{T} \right)^\mathsf{T}$$

so that the kernel is estimated by $z(A(P))$.

However, the projection coefficients of the p_λ functions do not have simple forms as before; instead, we must directly estimate the density as $\hat{p}$ using a technique such as kernel density estimation (KDE) and then estimate $\vec{a}(\hat{p}_\lambda)$ for each λ with numerical integration. Recall that the elements of $A(\hat{p})$ are of the form

$$a_\alpha\left(\hat{p}_{\lambda_j}^S\right) = \int_{[0,1]^d} \varphi_\alpha(t)\, \hat{p}_{\lambda_j}^S(t)\, \mathrm{d}t$$

where $j \in \{1,\dots,M\}$, $S \in \{R,I\}$, $\alpha \in V$. For small d, simple Monte Carlo integration is sufficient. Choosing $\{u_i\}_{i=1}^{n_e} \overset{iid}{\sim} \mathrm{Unif}\left([0,1]^\ell\right)$:

$$\hat{a}_\alpha\left(\hat{p}_{\lambda_j}^S\right) = \frac{1}{n_e} \sum_{i=1}^{n_e} \varphi_\alpha(u_i)\, \hat{p}_{\lambda_j}^S(u_i), \tag{4.10}$$

giving us an estimate of $A(\hat{p})$ which we call $\hat{A}(\hat{p})$.

In higher dimensions, three problems arise: (i) density estimation becomes statistically difficult, (ii) accurate numerical integration becomes expensive, and (iii) the embedding dimension increases exponentially. We can attempt to address (i) with sparse nonparametric graphical models (Lafferty et al. 2012) or other high-dimensional density estimation techniques (Sriperumbudur, Fukumizu, Kumar, et al. 2013). Point (ii) could be handled with MCMC integration; high-dimensional multimodal integrals remain particularly challenging to current MCMC techniques, but some progress is being made (e.g. Betancourt 2015; Lan et al. 2014 give a heuristic algorithm). Challenge (iii) requires some changes to the algorithm to address, as it does for Section 4.2.

Summary and Complexity The algorithm for computing random features $\{z(A(p_i))\}_{i=1}^N$ for the generalized RBF kernel based on an HDD ρ among a set of distributions $\{P_i\}_{i=1}^N$, given sample sets $\{X_i\}_{i=1}^N$ where $X_i = \{X_j^{(i)} \in [0,1]^d\}_{j=1}^{n_i} \overset{iid}{\sim} P_i$, is thus:

1. Draw M scalars $\lambda_j \overset{iid}{\sim} \frac{\mu}{Z}$ and $D/2$ vectors $\omega_r \overset{iid}{\sim} \mathcal{N}(0, \sigma^{-2} I_{2M|V|})$, in $O(M\,|V|\,D)$ time.
2. For each of the N input distributions i:

(a) Compute a KDE from X_i, $\hat{p}_i(u_j)$ for each u_j in (4.10), in $O(n_i n_e)$ time.

(b) Compute $\hat{A}(\hat{p}_i)$ using a numerical integration estimate as in (4.10), in $O(M\,|V|\,n_e)$ time.

(c) Get the random Fourier features, $z(\hat{A}(\hat{p}_i))$, in $O(M\,|V|\,D)$ time.

Supposing each $n_i \asymp n$, this process takes a total of $O\left(Nnn_e + NM\,|V|\,n_e + NM\,|V|\,D\right)$ time. Taking $|V|$ to be asymptotically $O(n)$, $n_e = O(D)$, and $M = O(1)$ for simplicity, this is $O(NnD)$ time, compared to about $O(N^2 n \log n + N^3)$ for using the k-NN estimator for divergences with corrections for indefiniteness, or $O(N^2n^2)$ for using the quadratic-time MMD estimator (as in Muandet, Schölkopf, et al. 2012).

4.3.1 Convergence bound

We bound the finite-sample error of our estimator for fixed densities p and q by considering each source of error: kernel density estimation ($\varepsilon_{\mathrm{KDE}}$); approximating $\mu(\lambda)$ with M samples (ε_λ); truncating the tails of the projection coefficients ($\varepsilon_{\mathrm{tail}}$); Monte Carlo integration ($\varepsilon_{\mathrm{int}}$); and the RKS embedding ($\varepsilon_{\mathrm{RKS}}$).

Proposition 4.10. *Fix p and q as two densities supported on $[0,1]^d$ satisfying some smoothness assumptions: that they are members of a periodic Hölder class $\Sigma_{per}(\beta, L_\beta)$ for some $\beta, L_\beta > 0$, that they are bounded below by ρ_* and above by ρ^*, and that their kernel density estimates are in $\Sigma_{per}(\hat{\gamma}, \widehat{L})$ for some $\hat{\gamma}, \widehat{L} > 0$ with probability at least $1-\delta$. Suppose we observe n samples from each.*

We will use the estimator of Section 4.3 with a suitable form of kernel density estimation to obtain a uniform error bound with a rate based on a function C^{-1} (Giné and Guillou 2002). We use the Fourier basis and choose $V = \{\alpha \in \mathbb{Z}^\ell \mid \sum_{j=1}^{\ell} |\alpha_j|^{2s} \le t\}$ for parameters $0 < s < \hat{\gamma}$, $t > 0$.

Then, for any $\varepsilon_{RKS} + \frac{1}{\sigma_k \sqrt{e}}\left(\varepsilon_{KDE} + \varepsilon_\lambda + \varepsilon_{tail} + \varepsilon_{int}\right) \le \varepsilon$:

$$\Pr\left(\left|K(p,q) - z(\hat{A}(\hat{p}))^\mathsf{T} z(\hat{A}(\hat{q}))\right| \ge \varepsilon\right) \le 2\exp\left(-D\varepsilon_{RKS}^2\right) + 2\exp\left(-M\varepsilon_\lambda^4/(8Z^2)\right) + \delta$$
$$+ 2C^{-1}\left(\frac{\varepsilon_{KDE}^4 n^{2\beta/(2\beta+d)}}{4\log n}\right) + 2M\left(1 - \mu\left([0, u_{tail}]\right)\right)$$
$$+ 8M\,|V|\exp\left(-\tfrac{1}{2}n_e\left(\frac{\sqrt{1+\varepsilon_{int}^2/(8\,|V|\,Z)} - 1}{\sqrt{\rho^*}+1}\right)^2\right)$$

where $u_{tail} := \sqrt{\max\left(0, \frac{\rho_* t}{8Md\widehat{L}^2}\frac{4^{\hat{\gamma}}-4^s}{4^{\hat{\gamma}}}\varepsilon_{tail}^2 - \frac{1}{4}\right)}$.

For a more detailed statement and the proof, see Appendix C.1.

The bound decreases when the function is smoother (larger β, $\hat{\gamma}$; smaller $\widehat{L}$) or lower-dimensional (d), or when we observe more samples (n). Using more projection coefficients (higher t or smaller s, giving higher $|V|$) improves the approximation but makes numerical integration more difficult. Likewise, taking more samples from μ (higher M) improves that approximation, but increases the number of functions to be approximated and numerically integrated.

4.3.2 Generalization to α-HDDs

Corollary 1 of Fuglede (2005), the core of our previous embedding, actually applies to a broader class of functions. Let an α-HDD be an HDD whose κ is α-homogeneous, in the sense that $\kappa(tx, ty) = t^\alpha \kappa(x, y)$. Thus the HDDs discussed previously are 1-HDDs. The embedding is just as before, except that

$$g_\lambda^{(\alpha)}(x) := \sqrt{Z} \frac{-\frac{1}{2}\alpha + \mathbb{i}\lambda}{\frac{1}{2}\alpha + \mathbb{i}\lambda} \left(x^{\frac{1}{2}\alpha + \mathbb{i}\lambda} - 1 \right), \tag{4.11}$$

and so of course the p_λ functions are altered accordingly as well. The equivalent of (4.8) is

$$\kappa(x, 1/x) = Z x^\alpha + Z x^{-\alpha} - 2 \int_{\mathbb{R}_{\geq 0}} \cos(2\lambda \log x) \, \mathrm{d}\mu(\lambda). \tag{4.12}$$

For example, L_2 is a 2-HDD defined by $\kappa(x, y) = (x - y)^2$; of course, κ is negative-definite. Note that, using (4.12), $\kappa(x, 1/x) = (x - 1/x)^2 = x^2 + x^{-2} - 2$ so that $\mu(\lambda) = \delta(\lambda = 0)$, and

$$g_0^{(2)}(x) := 1 - x,$$

so (using $M = 1$) the embedding (4.9) becomes simply

$$\rho^2(p, q) = \|(1 - p) - (1 - q)\|^2 + \|0 - 0\|^2 = \|p - q\|^2.$$

Proposition 4.10 could be extended to α-HDDs without too much difficulty.

4.3.3 Connection to MMD

2-HDDs are defined by, combining (4.7) and (4.11):

$$\begin{aligned} \rho^2(p, q) &= \int_{\mathcal{X}} \left[\int_{\mathbb{R}_{\geq 0}} \left| p(x)^{1+\mathbb{i}\lambda} - q(x)^{1+\mathbb{i}\lambda} \right|^2 \mathrm{d}\mu(\lambda) \right] \mathrm{d}x \\ &= \int_{\mathbb{R}_{\geq 0}} \int_{\mathcal{X}} \left| p(x)^{1+\mathbb{i}\lambda} - q(x)^{1+\mathbb{i}\lambda} \right|^2 \mathrm{d}x \, \mathrm{d}\mu(\lambda) \\ &= \int_{\mathbb{R}_{\geq 0}} \int_{\mathcal{X}} \left| p(x) e^{\mathbb{i}\lambda \log p(x)} - q(x) e^{\mathbb{i}\lambda \log q(x)} \right|^2 \mathrm{d}x \, \mathrm{d}\mu(\lambda). \end{aligned}$$

Meanwhile, Corollary 4(i) of Sriperumbudur, Gretton, et al. (2010) establishes that when k is a continuous shift-invariant kernel on $\mathcal{X} \subseteq \mathbb{R}^d$ and Ω the Fourier transform of $\underline{k}$:

$$\begin{aligned} \textsc{mmd}(P, Q)^2 &= \int_{\mathbb{R}^d} \left| \mathbb{E}_{X \sim P}[e^{\mathbb{i}\omega^\intercal X}] - \mathbb{E}_{Y \sim Q}[e^{\mathbb{i}\omega^\intercal Y}] \right|^2 \mathrm{d}\Omega(\omega) \\ &= \int_{\mathbb{R}^d} \left| \int_{\mathcal{X}} p(x) e^{\mathbb{i}\omega^\intercal x} \mathrm{d}x - \int_{\mathcal{X}} q(x) e^{\mathbb{i}\omega^\intercal x} \mathrm{d}x \right|^2 \mathrm{d}\Omega(\omega). \end{aligned}$$

This similarity in form is appealing, but a deeper connection between the two is elusive.

Chapter 5

Applications of distribution learning

We now turn to case studies in applying distributional kernels to real machine learning tasks:

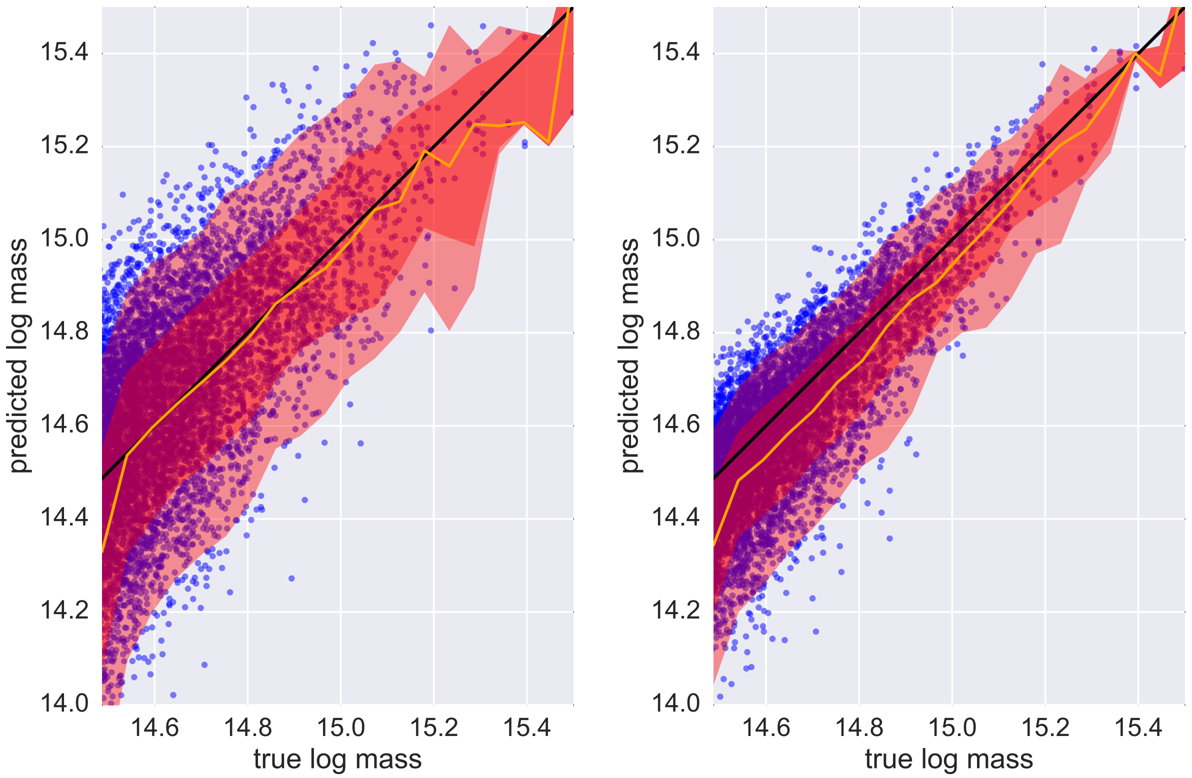
- Section 5.1 employs distribution regression to predict the total mass of galaxy clusters in observationally realistic settings. (Results previously published in Ntampaka, Trac, Sutherland, Battaglia, et al. 2015; Ntampaka, Trac, Sutherland, Fromenteau, et al. in press.)
- Section 5.2 examines the scalability of distribution embeddings on a synthetic problem of predicting the number of components in a Gaussian mixture (Sutherland, J. B. Oliva, et al. 2016).
- Section 5.3 studies scene recognition in natural images. Section 5.3.1 uses full-Gram matrix techniques with SIFT features (Póczos, Xiong, Sutherland, et al. 2012; Sutherland, Xiong, et al. 2012); Section 5.3.2 uses distribution embeddings with deep learning-derived features (Sutherland, J. B. Oliva, et al. 2016).
- Section 5.4 applies distribution regression to the photons observed by a small backpack-sized sensor to identify potentially harmful sources of radiation (Jin et al. 2016).

5.1 Dark matter halo mass prediction

Galaxy clusters are the most massive gravitationally bound system in the universe, containing up to hundreds of galaxies embedded in dark matter halos. Their properties, especially total mass, are extremely useful for making inferences about fundamental cosmological parameters, but because they are composed largely of dark matter, measuring that mass is difficult.

One classical method is that of Zwicky (1933). The virial theorem implies that the dispersion of velocities in a stable system should be approximately related to the halo mass as a power law; by measuring the Doppler shift of spectra from objects in the cluster, we can estimate the dispersion of velocities in the direction along our line of sight, and thus predict the total mass. Zwicky's estimate famously led him to the first formal inference about the presence of dark matter.

Experimental evidence, however, points towards various complicating factors that disturb this idealized relationship, and indeed results based on numerical simulation have shown that the predictions from this power law relationship are not as accurate as we would hope. We



(a) Power law results.

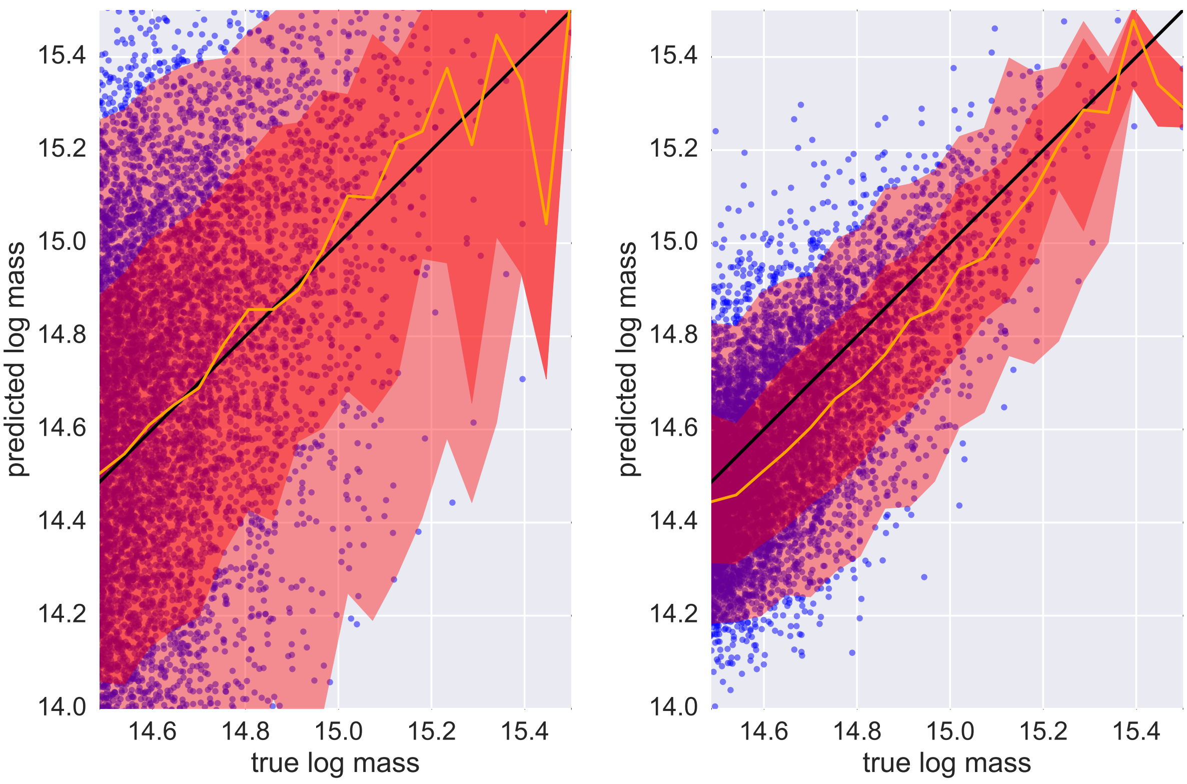
(b) SKL results on $|v_{\mathrm{los}}|$ features.

Figure 5.1: Performance for halo mass prediction, for power law (left) and distribution regression (right) approaches. Each test projection is plotted with its true log mass on the horizontal axis and prediction on the vertical axis. The black line shows perfect predictions; the yellow line gives the median of the predicted points, the darker red region shows 68% scatter, and the lighter red 95% scatter.

can therefore consider using all information available in the line-of-sight velocity distribution by directly learning a regression function from that distribution to total masses, based on data from simulation.

We assembled a catalog of massive halos from the MultiDark MDPL simulation (Klypin et al. 2014). The catalog contains 5 028 unique halos. Since we use only line-of-sight velocities, however, we can view each halo from multiple directions. For hyperparameter selection and testing, we use lines of sight corresponding to three perpendicular directions; for training, we additionally use projections sampled randomly from the unit sphere so as to oversample the rare high-mass halos. Different projections of the same halo are always assigned to the same fold for cross-validation. Ntampaka, Trac, Sutherland, Battaglia, et al. (2015) give a precise description of the details.

We then use the SKL estimator of Q. Wang et al. (2009) in a generalized RBF kernel on a simple one-dimensional feature set containing only the magnitude of the line-of-sight velocity. Figure 5.1 shows results, establishing that the distribution regression technique greatly outperforms the power law. The power law achieves a root mean squared error (RMSE) of 0.180, whereas the distribution learning method gets 0.118. Ntampaka, Trac, Sutherland, Battaglia, et al. (2015) also considered other featurizations, which performed similarly or sometimes slightly better, and has a much more thorough analysis of the results.



(a) Power law results.

(b) SKL results on $|v_{\text{los}}|$ features.

Figure 5.2: Performance for halo mass prediction with interlopers. Same format as Figure 5.1.

These results, however, differed from the true observational setting in one important way: we assumed perfect knowledge of cluster memberships. In actual observations, we would not know which objects belong to the cluster at hand, and which merely happen to appear nearby from our Earth-bound observation point. Standard practice for application of the power law-based approach is to employ complex systems for estimating which objects are gravitationally bound and which are not. Distribution regression with the SKL estimator, however, is far more robust to the presence of these interlopers than the power law approach. In Ntampaka, Trac, Sutherland, Fromenteau, et al. (in press), we modified the catalog to use a very simple heuristic for choosing the members of a cluster and then applied the same prediction techniques. The results are shown in Figure 5.2; the RMSE of the power law is now an enormous 0.434, where distribution regression is 0.177 — matching the performance of the power law predictions based on perfect knowledge about cluster membership.

5.2 Mixture estimation

Statistical inference procedures can be viewed as functions from distributions to the reals; we can therefore consider learning such procedures. Jitkrittum, Gretton, et al. (2015) trained MMD-based GP regression for the messages computed by numerical integration in an expectation propagation system, and saw substantial speedups by doing so. We, inspired by J. B. Oliva, Neiswanger, et al. (2014), consider a problem where we not only obtain speedups over traditional algorithms, but actually see superior results.

Specifically, we consider predicting the number of components in a Gaussian mixture. We generate mixtures as follows:

1. Draw the number of components Y_i for the ith distribution as $Y_i \sim \mathrm{Unif}\{1, \ldots, 10\}$.
2. For each component, select a mean $\mu_k^{(i)} \sim \mathrm{Unif}[-5, 5]^2$ and covariance $\Sigma_k^{(i)} = a_k^{(i)} A_k^{(i)} A_k^{(i)\mathsf{T}} + B_k^{(i)}$, where $a \sim \mathrm{Unif}[1, 4]$, $A_k^{(i)}(u, v) \sim \mathrm{Unif}[-1, 1]$, and $B_k^{(i)}$ is a diagonal 2×2 matrix with $B_k^{(i)}(u, u) \sim \mathrm{Unif}[0, 1]$.
3. Draw a sample $X^{(i)}$ from the equally-weighted mixture of these components.

An example distribution and sample from it is shown in Figure 5.3; predicting the number of components is difficult even for humans.

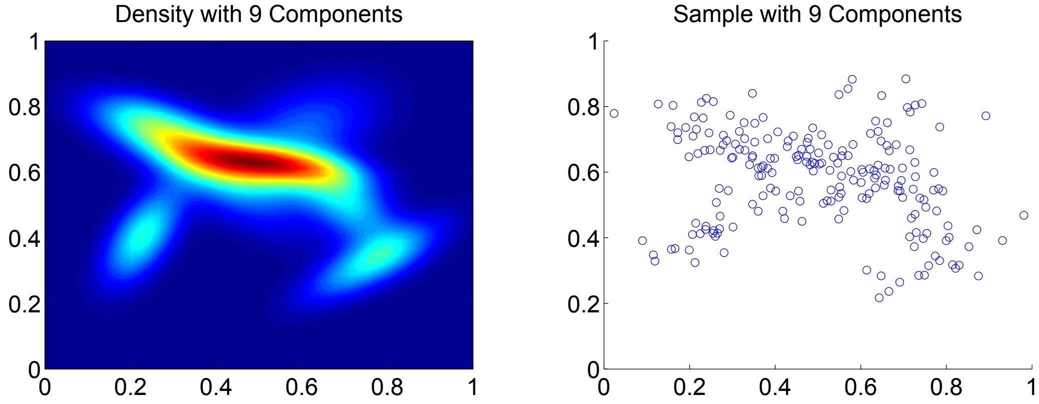
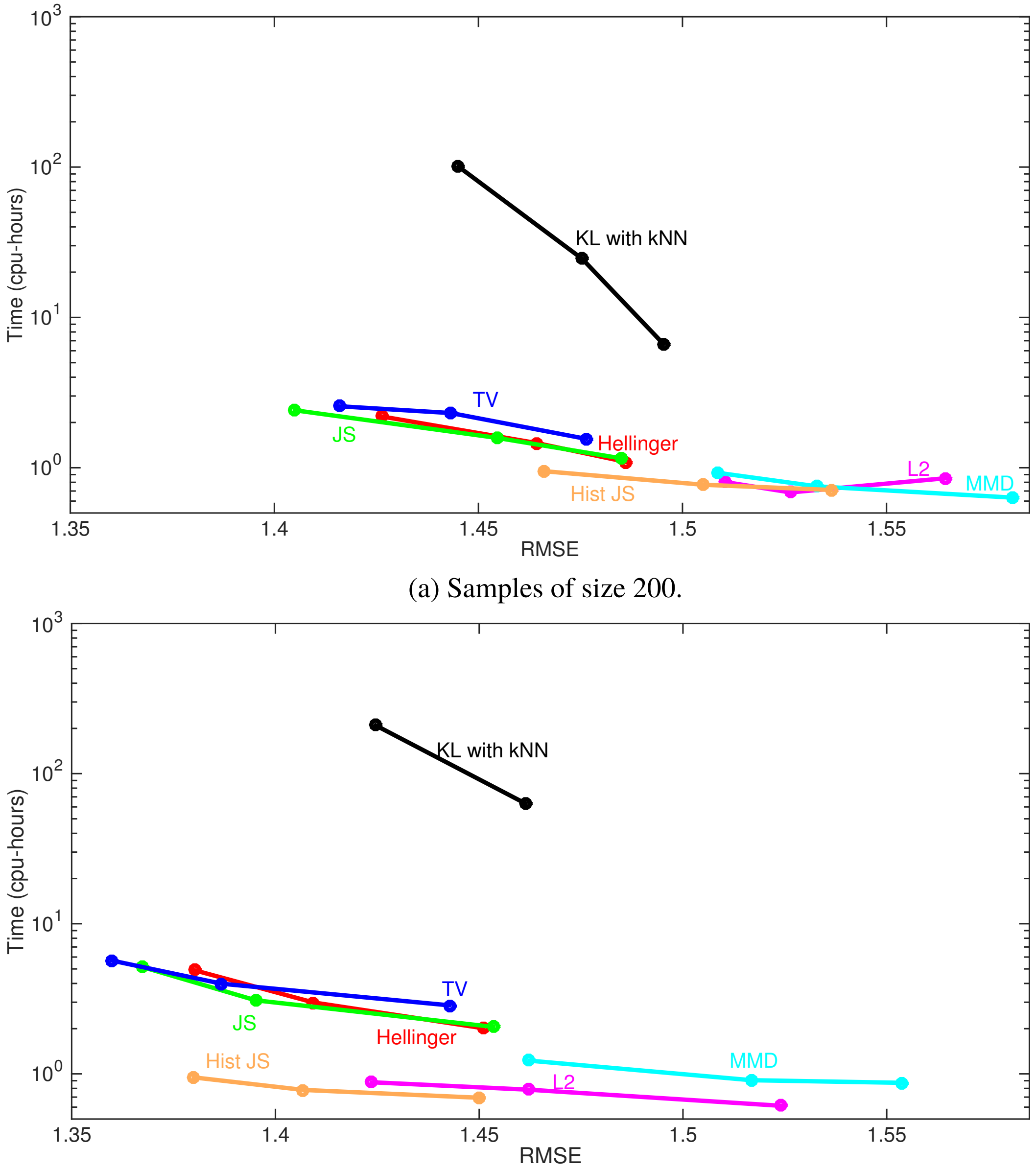


Figure 5.3: Example of a mixture with 9 components and a sample from it of size $n = 200$.

We compare generalized RBF kernels based on the MMD, L_2, and HDD embeddings of Chapter 4 as well as the JS embedding of Vedaldi and Zisserman (2012) and the full Gram matrix techniques of Section 2.4 applied to the SKL estimator of Q. Wang et al. (2009).

Figure 5.4 presents results for predicting with ridge regression the number of mixture components Y_i, given a varying number of sample sets X_i, with $|X_i| \in \{200, 800\}$; we use $D = 5\,000$. The HDD-based kernels achieve substantially lower error than the L_2 and MMD kernels in both cases. They also outperform the histogram kernels, especially with $|X_i| = 200$, and the KL kernel. Note that fitting mixtures with EM and selecting a number of components using AIC (Akiake 1973) or BIC (Schwarz 1978) performed much worse than regression; only AIC with $|X_i| = 800$ outperformed a constant predictor of 5.5. Linear versions of the L_2 and MMD kernels, based on (2.2) instead of the (2.3) results shown, were also no better than the constant predictor.



(a) Samples of size 200.

(b) Samples of size 800.

Figure 5.4: Error and computation time for estimating the number of mixture components. The three points on each line correspond to training set sizes of 4K, 8K, and 16K; error is on the fixed test set of size 2K. Note the logarithmic scale on the time axis. The KL kernel for sets of size 800 and 16K training sets was too slow to run. AIC-based predictions achieved RMSEs of 2.7 (for 200 samples) and 2.3 (for 800); BIC errors were 3.8 and 2.7; a constant predictor of 5.5 had RMSE of 2.8.

The HDD embeddings were more computationally expensive than the other embeddings, but much less expensive than the KL kernel, which grows at least quadratically in the number of distributions. Note that the histogram embeddings used an optimized C implementation by the paper's authors (Vedaldi and Fulkerson 2008), and the KL kernel used the optimized implementation of `skl-groups`, whereas the HDD embeddings used a simple Matlab implementation.

5.3 Scene recognition

Representing images as a collection of local patches has a long and successful history in computer vision.

5.3.1 SIFT features

The traditional approach selects a grid of patches, computes a hand-designed feature vector such as SIFT (Lowe 2004) for each patch, possibly appends information about the location of the patch, and then uses the BOW representation for this set of features. We will first consider the use of distributional distance kernels for this feature representation.

We present here results on the 8-class OT scene recognition dataset (A. Oliva and Torralba 2001); the original papers show results on additional image datasets. This dataset contains 8 outdoor scene categories, illustrated in Figure 5.5. There are 2 688 total images, each about 256×256 pixels.

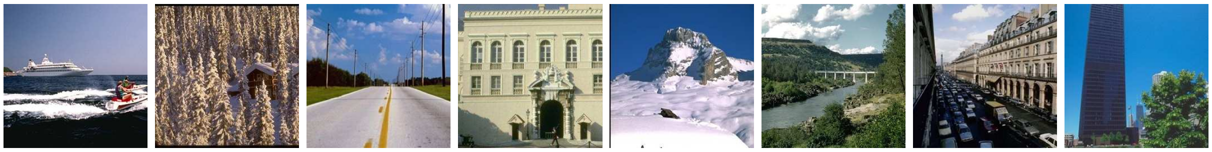


Figure 5.5: The 8 OT categories: *coast, forest, highway, inside city, mountain, open country, street, tall building*.

We extracted dense color SIFT features (Bosch et al. 2008) at six different bin sizes using VLFEAT (Vedaldi and Fulkerson 2008), resulting in about 1 815 feature vectors per image, each of dimension 384. We used PCA to reduce these to 53 dimensions, preserving 70% of the variance, appended relative y coordinates, and standardized each dimension. (The paper contains precise details.)

The results of 10 repeats of 10-fold cross-validation are shown in Figure 5.6. Each approach uses a generalized RBF kernel. Here BOW refers to vector quantization with k-means ($k = 1\,000$), PLSA to the approach of Bosch et al. (2006), G-KL and G-PPK to the KL and Hellinger divergences between Gaussians fit to the data, GMM-KL to the KL between Gaussian mixtures fit to the data with expectation maximization (computing via Monte Carlo), PMK to the pyramid matching kernel of Grauman and Darrell (2007), MMK to the MMK with a Gaussian base kernel, NPH to the nonparametric Hellinger estimate of Póczos, Xiong, Sutherland, et al. (2012), and NPR- to the R_α estimates. The horizontal line shows the best previously reported result (Qin and Yung 2010), though others have since slightly surpassed our results here.

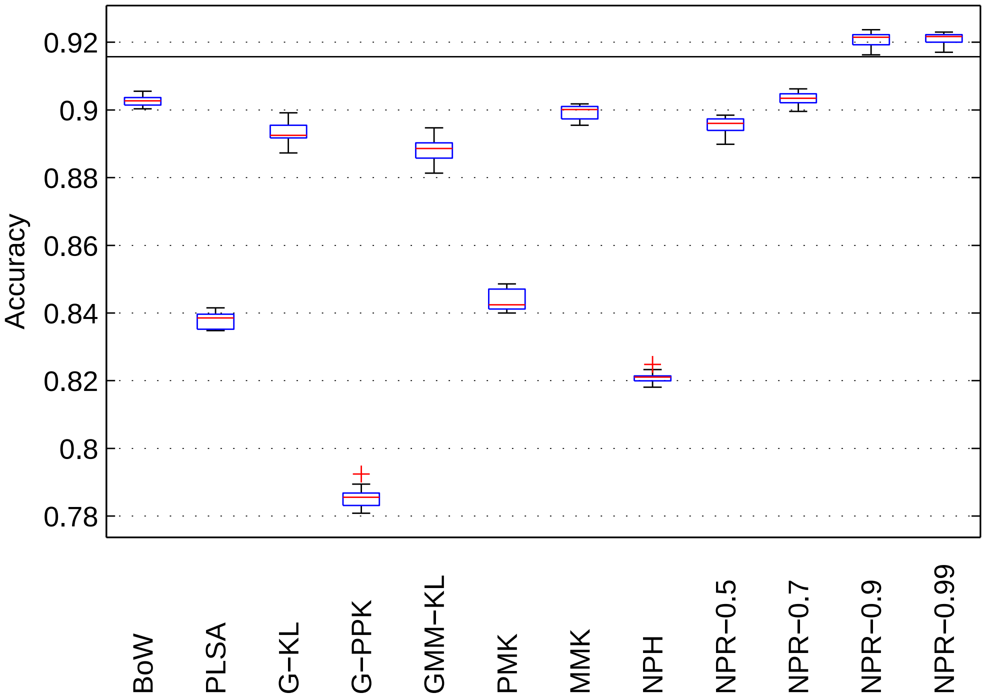


Figure 5.6: Accuracies on the OT dataset.

5.3.2 Deep features

For the last several years, however, modern computer vision has become overwhelmingly based on deep neural networks. Image classification networks typically broadly follow the architecture of Krizhevsky et al. (2012), i.e. several convolutional and pooling layers to extract complex features of input images followed by one or two fully-connected layers to classify the images.

The activations are of shape $n \times h \times w$, where n is the number of filters; each unit corresponds to an overlapping patch of the original image. We can therefore treat the activations as a sample of size hw from an n-dimensional distribution. Wu et al. (2016) set accuracy records on several scene classification datasets with a particular method of extracting features from distributions. That method, however, resorts to ad-hoc statistics; we compare to our more principled alternatives here.

We consider here the Scene-15 dataset (Lazebnik et al. 2006), which contains 4 485 natural images in 15 categories based on location. (It is a superset of the OT dataset previously considered, but is available only in grayscale.) We follow Wu et al. (2016) in extracting features from the last convolutional layer of the `imagenet-vgg-verydeep-16` model (Simonyan and Zisserman 2015). We replace that layer's rectified linear activations with sigmoid squashing to $[0, 1]$.[1] After resizing the images as did Wu et al. (2016), hw ranges from 400 to 1 000. There are 512 filter dimensions; we concatenate features $\hat{A}(\hat{p}_i)$ extracted from each independently.

We select 100 images from each class for training, and test on the remainder; Figure 5.7 shows the results of 10 random splits. We do not add any spatial information to the model, unlike

[1] We used piecewise-linear weights such that 0 maps to 0.5, the 90th percentile of the positive observations maps to 0.9, and the 10th percentile of the negative observations to 0.1, for each filter.

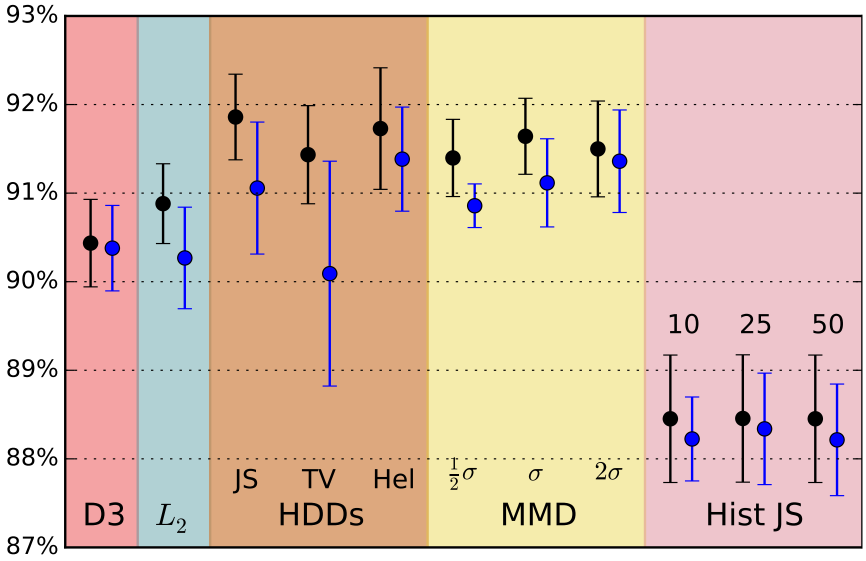


Figure 5.7: Mean and standard deviation accuracies on the Scene-15 dataset. The left, black lines show performance with linear features; the right, blue lines show generalized RBF embedding features. D3 refers to the method of Wu et al. (2016). MMD bandwidths are relative to σ, the median of pairwise distances; histogram methods use varying numbers of bins.

Wu et al. (2016); still, we match the best prior published performance of 91.59 ± 0.48, using a deep network trained on a large scene classification dataset (Zhou et al. 2014). Adding spatial information brought the D3 method of Wu et al. (2016) slightly above 92% accuracy; their best hybrid method obtained 92.9%. Using these features, however, our methods match or beat MMD and substantially outperform D3, L_2, and the histogram embeddings.

5.4 Small-sensor detection of radiation sources

Preventing the proliferation of nuclear weapons and stopping nuclear terrorist attacks is one of the prime responsibilities of security agencies. Tactical nuclear weapons are very portable and pose great risks to urban environments. Radioactive isotopes stolen from medical uses pose another threat. Although certain border check points can afford to require potential threats to go through large and expensive detectors, mobile radiation detectors are vital for finding radiation sources which either successfully passed through those choke points or managed to avoid them. In certain situations, sensors carried by pedestrians in a backpack are a promising tactic for seeking out these sources. Much of the time, however, the targets are relatively weak, potentially shielded, and masked by highly-variable patterns of background radiation, especially in the cluttered urban environments where dirty bombs or improperly stored radioactive material can cause the most harm. We need, therefore, sophisticated systems which can detect radioactive sources in real time while also maintaining a low false alarm rate.

With small sensors such as those considered here, the strong Compton effect makes observations of a photon's energy very noisy: high-energy photons are often measured as if they were much lower-energy. Combined with the fewer total photons received by the smaller sensor, this means that existing detection algorithms are typically outperformed in this setting by a simple threshold on the total number of photons observed, regardless of energy.

Instead, we can invert a probabilistic sensor response model to obtain a distribution of possible energies corresponding to each photon we observe. Given such a model, we use a simple Monte Carlo technique: replace each of the n observed photon energies with 1 000 samples from the distribution of possible true energies corresponding to the observed photon. We then model the background distribution of radiation: for a given source of radiation, say Cs137, we pick certain ranges of energy corresponding to that source (based on the signal-to-noise ratio compared to typical background distributions). Then we model the expected behavior within those energies by performing distribution regression from the *other* energy levels to the total number of photons received in the high-SNR energy levels. That is, we predict a total count $\hat{y}$ of photons in the high-SNR energy levels based on the distribution of photon energies observed at all other energy levels. The likelihood of source presence is then determined by the departure from the prediction: $\frac{y-\hat{y}}{\sqrt{\hat{y}}}$, where y is the observed number of photons in those regions and $\hat{y}$ the prediction.

We simulated this process by taking background data from the RadMAP dataset (Quiter et al. 2015), which comprises four hours of observations of the MISTI mobile detection vehicle (Mitchell et al. 2009) in the Berkeley, CA area. We also obtained characterizations of source spectra from collaborators Simon Labov and Karl Nelson at Lawrence Livermore. We generated background data from the observations made in the relatively large-sensor RadMAP data by simulating small-sensor measurements of it; we trained on background data, then evaluated on both distinct background samples and samples where observations corresponding to the source were injected. Details are given by Jin (2016).

Figure 5.8 shows results for detecting Cs137 sources at various classification thresholds. At low false alarm rates, the most relevant regime since true sources are hopefully quite rare in practice, the distribution regression method substantially outperforms the total counts algorithm; as the false alarm rate is allowed to increase, total counts catches up but never outperforms distribution regression.

Figure 5.9 shows the improvement in probability of detection over total counts at the 10^{-3} false alarm rate across 40 different sources. The majority of sources are better-detected by distribution regression than by total counts, some of them substantially so. Jin (2016) shows that distribution regression performs better in cases where the source's energy output is more concentrated in certain energy levels, as might be expected. He also shows that the improvement is consistent across different experimental setups, corresponding to varying the strength of the source and the size of the sensor.

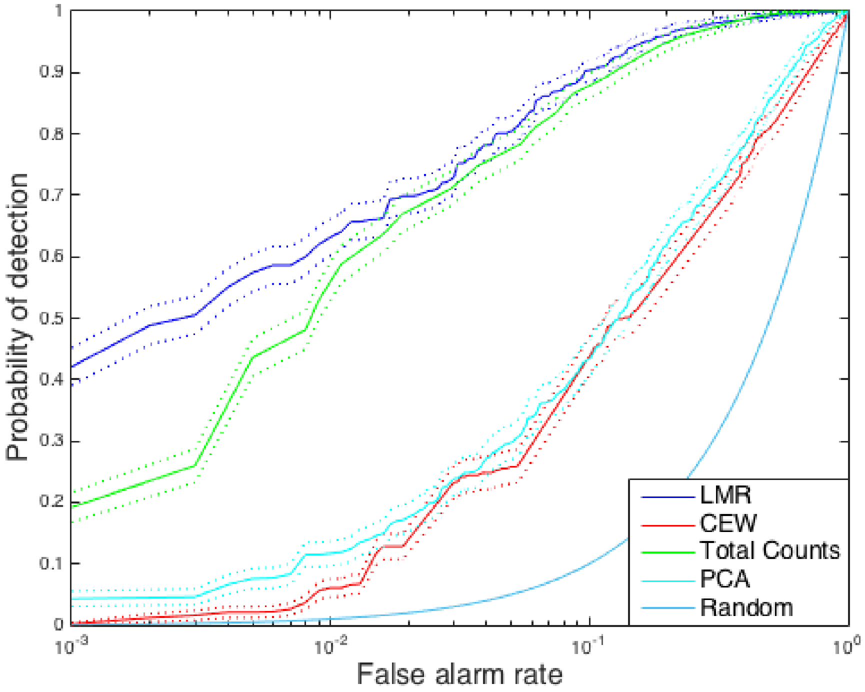


Figure 5.8: Receiver operating characteristic for different detection methods, with log-scale for the false alarm rate. LMR refers to *list mode regression*, the distributional regression technique; CEW is *censored energy windowing*; PCA to background subtraction via principal components analysis; random shows the hypothetical performance of a random classifier.

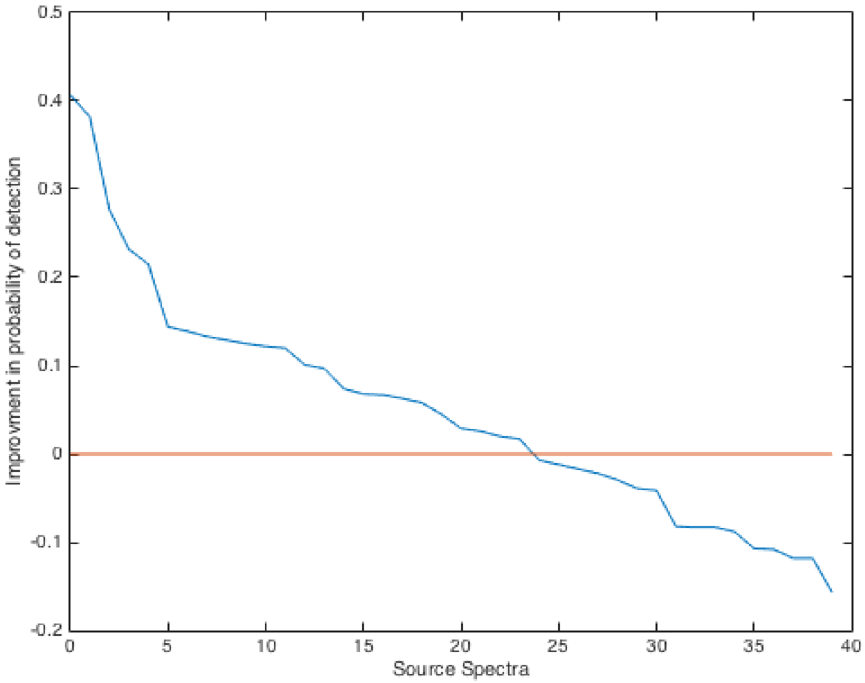


Figure 5.9: Pairwise improvement in probability of detection at false alarm rate 0.001 for 40 different sources, sorted by the improvement.

Chapter 6

Choosing kernels for hypothesis tests

So far, we have assumed a particular kernel $k : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ and used it to learn a classification function $f : \mathcal{P} \to \{-1, 1\}$ or a regression function $f : \mathcal{P} \to \mathbb{R}$. Although several of the kernels we have studied give good empirical performance on a variety of problems, with complex forms of data we must first choose an appropriate feature extraction pipeline (such as the SIFT or deep network features for the images in Section 5.3). Even in simple situations, we often have a family of kernels we expect to work well but must pick an element of that family, e.g. bandwidth selection in Gaussian RBF kernels. The field of kernel learning is an extensively studied but challenging approach to this form of problem (Gönen and Alpaydın 2011; Z. Yang et al. 2015; J. B. Oliva, Dubey, et al. 2016; Wilson et al. 2016). Though we could adapt those methods to distributional settings, we will instead study here the related problem of *two-sample testing*.

Specifically, we observe samples $X = \{x_1, \dots, x_m\} \sim P^m$ and $Y = \{y_1, \dots, y_m\} \sim Q^m$.[1] We wish to test the null hypothesis $H_0 : P = Q$ versus the alternative $H_1 : P \neq Q$. This problem has many important applications including independence testing (Gretton, Bousquet, et al. 2005), feature selection (Song et al. 2012), modeling of neuroimaging results (Tao and Feng 2016), data integration and automated attribute matching (Gretton, Borgwardt, et al. 2012), and guiding the training of generative models (Dziugaite et al. 2015; Y. Li et al. 2015).The problem is connected to but in some senses easier than training a classifier to distinguish P from Q (Sriperumbudur, Fukumizu, Gretton, Lanckriet, et al. 2009).

One standard approach to performing these tests is to choose a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and then use a test statistic based on an estimate of MMD_k between the samples. In the standard hypothesis testing framework, we choose a threshold c_α as the $(1 - \alpha)$th quantile of the distribution of the test statistic under H_0, and reject H_0 if the statistic exceeds the threshold.

Many kernels, including the Gaussian RBF, are *characteristic* (Fukumizu et al. 2008), implying that these tests are consistent: as $m \to \infty$, the power (that is, the probability that we reject H_0 when H_1 holds) converges to 1. Thus, given unlimited data and computational budget, we can choose k as an arbitrary characteristic kernel. In practice, however, the power of the test depends greatly on the choice of kernel. For example, if we select k from the family of Gaussian RBF kernels, a bandwidth too different from the scale on which P and Q differ will be unable to efficiently detect those differences. We thus need a criterion with which to select a kernel from

[1] For simplicity, we assume here that the sample sizes are equal. The unequal case would not be fundamentally more difficult.

some family.

Moreover, in high dimensions, even detecting shifts in the means of distributions becomes very difficult with general-purpose kernels (Ramdas, Reddi, et al. 2015). In structured domains like images, simple kernels like the Gaussian RBF also approximate natural notions of similarity quite poorly except when complex featurizations are first applied; this was the problem encountered by, for example, Dziugaite et al. (2015). Thus, we would like to be able to choose complex kernels capable of examining the distributions in ways particular to the domain at hand.

In this chapter, we develop a criterion for estimating the power of a kernel k on a particular two-sample test. This criterion is differentiable, so that we can optimize it even when we use complex structures such as deep networks within the kernel.

6.1 Estimators of MMD

Before discussing the kernel choice criterion and its antecedents, we need to briefly discuss some different choices of estimators for MMD.

Pairwise estimators Perhaps the simplest estimator for MMD is as follows:

$$\widehat{\text{MMD}}_b^2(X, Y) := \frac{1}{m^2} \sum_{i=1}^m \sum_{j=1}^m k(X_i, X_j) + \frac{1}{m^2} \sum_{i=1}^m \sum_{j=1}^m k(Y_i, Y_j) - \frac{2}{m^2} \sum_{i=1}^m \sum_{j=1}^m k(X_i, Y_j).$$

This is the exact MMD between the empirical distributions of the samples X and Y. Note, however, that the first two sums include terms of the form $k(X_i, X_i)$; it turns out these bias the estimator upwards. If we remove them, we get the minimum variance unbiased estimator (Gretton, Borgwardt, et al. 2012):

$$\widehat{\text{MMD}}_u^2(X, Y) := \frac{1}{\binom{m}{2}} \sum_{i=1}^m \sum_{j>i} k(X_i, X_j) + \frac{1}{\binom{m}{2}} \sum_{i=1}^m \sum_{j>i} k(Y_i, Y_j) - \frac{2}{m^2} \sum_{i=1}^m \sum_{j=1}^m k(X_i, Y_j).$$

The following estimator is very similar: it has slightly higher variance, by ignoring terms of the form $k(X_i, Y_i)$, but allows us to apply the theory of U-statistics (Serfling 1980, Chapter 5) more directly to the estimator. Let $W_i := (X_i, Y_i)$. Then:

$$\begin{aligned} h(w, w') &:= k(x, x') + k(y, y') - k(x, y') - k(y', x) \qquad (6.1) \\ \widehat{\text{MMD}}_U^2(X, Y) &:= \frac{1}{\binom{m}{2}} \sum_{i \neq j} h(W_i, W_j). \end{aligned}$$

These estimators are sometimes referred to as the "quadratic-time" estimators, because they take $O(m^2)$ time to evaluate. It takes $O(m)$ memory, because all samples must be stored in memory.

Streaming estimators When we have a very large, perhaps unbounded, number of samples available and wish to perform the best test available with a given computational budget, or perhaps when performing the test under strict memory restrictions, the following streaming estimator is useful. Assume for the sake of convenience that m is even.

$$\widehat{\text{MMD}}_s^2(X, Y) := \frac{2}{m} \sum_{i=1,3,5,\ldots,m-1} h(W_i, W_{i+1})$$

where we again use the h function from (6.1). Thus, we examine pairs of inputs at a time, and once we have evaluated one pair we can forget it and move onto the next.

This estimator is useful in the streaming setting, and convenient to analyze because its terms are independent. It takes $O(m)$ time to compute, and $O(1)$ memory. They are sometimes referred to as the "linear-time estimators"; we avoid that term, however, because the embedding-based estimators also take linear time.

When m is limited, however, the streaming estimator is far less efficient than the pairwise estimators. Ramdas, Reddi, et al. (2015) show that even for testing against mean-shift alternatives, the asymptotic power in the low-signal-to-noise, high-dimensional regime behaves like $\Phi(\alpha m)$ for the pairwise estimator and $\Phi(\alpha \sqrt{m})$ for the streaming estimator, so that approximately m^2 samples are needed for the streaming estimator to have equivalent power to the quadratic estimator.

Embedding-based estimators This is the approach of Section 4.1: assuming we have an approximate embedding $k(x, y) \approx z(x)^\mathsf{T} z(y)$, and letting $\bar{z}(X) = \frac{1}{m} \sum_{i=1}^m z(X_i)$, we simply have

$$\widehat{\text{MMD}}_b^2(X, Y) \approx \lVert \bar{z}(X) - \bar{z}(Y) \rVert^2 .$$

We can also approximate the unbiased estimator, though it is not nearly as nice:

$$\widehat{\text{MMK}}_u^2(X, X) \approx \frac{m^2}{m(m-1)} \left(\lVert \bar{z}(X) \rVert^2 - \frac{1}{m^2} \sum_{i=1}^m \lVert z(X_i) \rVert^2 \right)$$
$$\widehat{\text{MMD}}_u^2(X, Y) \approx \widehat{\text{MMK}}_u^2(X, X) + \widehat{\text{MMK}}_u^2(Y, Y) - 2 \bar{z}(X)^\mathsf{T} \bar{z}(Y).$$

When $\lVert z(x) \rVert = 1$, as with the shift-invariant embedding $\tilde{z}$ of (3.1), $\widehat{\text{MMK}}_u^2(X, X)$ simplifies to $\frac{m}{m-1} \lVert \bar{z}(X) \rVert^2 - \frac{1}{m-1}$. For the non-shift-invariant embedding $\breve{z}$ (3.2), this is true only in expectation.

We could similarly approximate $\widehat{\text{MMD}}_U^2$ if we wished, by subtracting off the terms corresponding to $z(X_i)^\mathsf{T} z(Y_i)$.

Chwialkowski et al. (2015) studied the performance of two-sample tests using these estimators, and found that although their performance can be surprisingly poor in certain situations (their Proposition 1), a related class of tests using similar embeddings performs well.

6.2 Estimators of the variance of $\widehat{\text{MMD}}^2$

Some of the kernel choice criteria we will develop shortly will require estimates of the variance of $\widehat{\text{MMD}}^2$.

Streaming estimator The asymptotic distribution for $\widehat{\text{MMD}}_s^2$ is simple: because it is an average of independent random variables the central limit theorem tells us that under either the null or the alternative,

$$\frac{\widehat{\text{MMD}}_s^2 - \text{MMD}^2}{\sqrt{V_s^{(m)}}} \xrightarrow{D} \mathcal{N}(0, 1) \tag{6.2}$$

where

$$V_s^{(m)} := \frac{m}{2} \left(\mathbb{E}_{w,w'}\, h^2(w, w') - \left[\mathbb{E}_{w,w'}\, h(w, w') \right]^2 \right). \tag{6.3}$$

This can be estimated in a streaming fashion as:

$$\widehat{V}_s^{(m)} := \frac{4}{m} \sum_{i=1,5,9,\ldots,m-3} \left(h(W_i, W_{i+1}) - h(W_{i+2}, W_{i+3}) \right)^2. \tag{6.4}$$

U-estimator The asymptotic distribution for $\widehat{\text{MMD}}_U^2$ is complex under H_0, and we will resort to permutation tests to determine the test threshold. Under H_1, however, $\widehat{\text{MMD}}_U^2$ is asymptotically normal (Gretton, Borgwardt, et al. 2012):

$$\frac{1}{\sqrt{V_U^{(m)}}} \left(\widehat{\text{MMD}}_U^2 - \text{MMD}^2 \right) \xrightarrow{D} \mathcal{N}(0, 1), \tag{6.5}$$

with

$$V_U^{(m)} := \frac{4(m-2)}{m(m-1)} \zeta_1 + \frac{2}{m(m-1)} \zeta_2, \tag{6.6}$$

where $\zeta_1 := \operatorname{Var}_v[\mathbb{E}_{v'}[h(v, v')]]$ and $\zeta_2 := \operatorname{Var}_{v,v'}[h(v, v')]$. This is established for U-statistics in general by Serfling (1980, Chapter 5); the analysis here was partially carried out for MMD in particular in Appendix A of Bounliphone et al. (2015). Using φ to denote the feature map of the kernel k and $\mu_x = \mathbb{E}_x\, \varphi(x)$, $\mu_y = \mathbb{E}_y\, \varphi(y)$, we have that:

$$\begin{aligned} \zeta_1 &= \mathbb{E}_v \left[\mathbb{E}_{v'}[h(v, v')]^2 \right] - \text{MMD}^2 \\ &= \mathbb{E}_{x,y} \left[\left(\langle \varphi(x), \mu_x \rangle + \langle \varphi(y), \mu_y \rangle - \langle \varphi(x), \mu_y \rangle - \langle \mu_x, \varphi(y) \rangle \right)^2 \right] - \text{MMD}^2. \end{aligned}$$

Expanding the square, we get an (unpleasant) expression in terms of expectations. ζ_2 can be calculated similarly.

We can estimate these terms based on a sample as follows. Let $K_{XX} := \left[k(X_i, X_j) \right]_{ij}$, $K_{YY} := \left[k(Y_i, Y_j) \right]_{ij}$, $K_{XY} := \left[k(X_i, Y_j) \right]_{ij}$, and $\mathbf{1}$ refer to the all-ones vector of length m. Let $\tilde{K}_{XX}$, $\tilde{K}_{YY}$, $\tilde{K}_{XY}$ be the kernel matrices with diagonal elements set to 0. Let $\|\cdot\|_F$ denote the Frobenius norm. Then:

$$
\begin{aligned}
\hat{\zeta}_1 = \frac{1}{m(m-1)(m-2)} \left(\mathbf{1}^\mathsf{T} \tilde{K}_{XX} \tilde{K}_{XX} \mathbf{1} - \|\tilde{K}_{XX}\|_F^2 \right) - \left(\frac{1}{m(m-1)} \mathbf{1}^\mathsf{T} \tilde{K}_{XX} \mathbf{1} \right)^2 \\
- \frac{2}{m^2(m-1)} \mathbf{1}^\mathsf{T} \tilde{K}_{XX} K_{XY} \mathbf{1} + \frac{2}{m^3(m-1)} \mathbf{1}^\mathsf{T} \tilde{K}_{XX} \mathbf{1} \mathbf{1}^\mathsf{T} K_{XY} \mathbf{1} \\
+ \frac{1}{m(m-1)(m-2)} \left(\mathbf{1}^\mathsf{T} \tilde{K}_{YY} \tilde{K}_{YY} \mathbf{1} - \|\tilde{K}_{YY}\|_F^2 \right) - \left(\frac{1}{m(m-1)} \mathbf{1}^\mathsf{T} \tilde{K}_{YY} \mathbf{1} \right)^2 \\
- \frac{2}{m^2(m-1)} \mathbf{1}^\mathsf{T} \tilde{K}_{YY} K_{XY}^\mathsf{T} \mathbf{1} + \frac{2}{m^3(m-1)} \mathbf{1}^\mathsf{T} \tilde{K}_{YY} \mathbf{1} \mathbf{1}^\mathsf{T} K_{XY} \mathbf{1} \\
+ \frac{1}{m^2(m-1)} \left(\mathbf{1}^\mathsf{T} K_{XY}^\mathsf{T} K_{XY} \mathbf{1} - \|K_{XY}\|_F^2 \right) - 2 \left(\frac{1}{m^2} \mathbf{1}^\mathsf{T} K_{XY} \mathbf{1} \right)^2 + \frac{1}{m^2(m-1)} \left(\mathbf{1}^\mathsf{T} K_{XY} K_{XY}^T \mathbf{1} - \|K_{XY}\|_F^2 \right),
\end{aligned}
$$

and

$$
\hat{\zeta}_2 = \frac{1}{m(m-1)} \left\| \tilde{K}_{XX} + \tilde{K}_{YY} - \tilde{K}_{XY} - \tilde{K}_{XY}^\mathsf{T} \right\|_F^2 .
$$

We then define

$$
\widehat{V}_U^{(m)} := \begin{cases} \frac{4(m-2)}{m(m-1)} \hat{\zeta}_1 + \frac{2}{m(m-1)} \hat{\zeta}_2 & \text{when } \frac{4(m-2)}{m(m-1)} \hat{\zeta}_1 + \frac{2}{m(m-1)} \hat{\zeta}_2 > 0 \\ \frac{2}{m(m-1)} \hat{\zeta}_2 & \text{otherwise} \end{cases} . \tag{6.7}
$$

6.3 MMD kernel choice criteria

We now suppose that we have some class of kernels $\mathcal{K}$, and would like to choose an element $k \in \mathcal{K}$ with which to conduct our test.

In general, we will divide the observed data X and Y into two partitions: one “training sample” to choose the kernel, and one “testing sample” to evaluate the test. Doing so loses some statistical power, but the test statistic distribution becomes quite complicated when the kernel can depend on the data.

6.3.1 Median heuristic

Perhaps the most common criterion for choosing k applies only to the case where $\mathcal{K}$ is the family of Gaussian RBF kernels with different bandwidths. This heuristic proposes to set σ to the median pairwise distance in the joint sample $X \cup Y$; despite its simplicity, it performs well on many problems.

Reddi et al. (2014) studied its theoretical performance in high-dimensional problems; Ramdas, Reddi, et al. (2015) provide some theoretical justification in the particular case of testing for mean-difference alternatives in the high-dimensional regime.

6.3.2 Marginal likelihood maximization

Flaxman, Sejdinovic, et al. (2016) propose a Bayesian model for learning kernel embeddings, effectively adding a Gaussian Process prior to the estimator of the mean embedding. Doing so allows for a fully-Bayesian treatment of learning the corresponding kernel, and they give an example of using learned kernels on testing problems. The kernel selection criterion, however, is fully unsupervised: it can only give the kernel choice that best describes the joint data, not one that best distinguishes between the two datasets. In this respect, it is somewhat similar to the median heuristic, though it has some ability to recognize when the data vary on multiple scales.

6.3.3 Maximizing MMD

Sriperumbudur, Fukumizu, Gretton, Lanckriet, et al. (2009) proposed choosing the kernel k which maximizes $\textsc{mmd}_k(X, Y)$. They showed that, for certain classes of kernels $\mathcal{K}$, the resulting test is consistent; additionally, it performs well empirically on many problems.

It does not directly optimize the test power, however: increasing the MMD estimate often also increases its variance and thus the required test threshold to exceed.

6.3.4 Cross-validation of loss

Gretton, Sriperumbudur, et al. (2012) propose as a method of comparison to choose kernel values via cross-validation, following Sugiyama et al. (2011), from the "classifier" perspective.

First, Sriperumbudur, Fukumizu, Gretton, Lanckriet, et al. (2009) establishes the following interpretation of MMD as a classifier: first, define the *witness function* as $f := \mu_P - \mu_Q$, i.e.

$$f(t) := \frac{1}{m}\sum_{i=1}^{m} k(x_i, t) - \frac{1}{m}\sum_{i=1}^{m} k(y_i, t).$$

Note that $f := \arg\sup_{f' \in \mathcal{H}_k} \mathbb{E}_{X\sim P}\, f'(X) - \mathbb{E}_{Y\sim Q}\, f'(Y)$, using the definiton of MMD as an integral probability metric. Then, we can view $\operatorname{sign}(f)$ as a Parzen window classifier trained with the points from X as positives and from Y as negatives. The MMD is then the negation of the linear loss function for that classifier.

Following this view of MMD, one can choose a kernel k by choosing the best Parzen window classifier via cross-validation. That is, divide the data into K folds, and then for each fold, learn a witness function f on the other $K - 1$ folds and evaluate its linear loss on the remaining fold. Optionally, repeat this process for several splits. Choose the kernel with the lowest linear loss.

This process requires evaluating each training set against the validation set, so that even when the streaming estimator is used, quadratically many comparisons must be made.

This method is actually quite similar to choosing the kernel via maximizing the MMD, but with a cross-validated estimate of MMD rather than evaluating on only one set. Strathmann (2012) found that, in certain problems, this approach outperformed maximizing the MMD.

6.3.5 Cross-validation of power

Strathmann (2012) proposes another method for applying cross-validation to kernel choice: directly estimate the power via cross-validation. Split the data into K folds, repeatedly performing a two-sample test on $K - 1$ of the folds (and ignoring the other fold). Repeat the data-splitting process. Then, choose the kernel which rejected the null distribution most often.

This approach can be performed in linear time when using $\widehat{\textsc{mmd}}_s^2$, and was found by Strathmann (2012) to outperform cross-validation based on the loss, and sometimes the t-statistic approach discussed shortly, in the streaming setting.

6.3.6 Embedding-based Hotelling stastistic

Jitkrittum, Szabó, et al. (2016) showed that one can perform kernel selection in the tests of Chwialkowski et al. (2015) simply by maximizing the test statistic.

6.3.7 Streaming t-statistic

Gretton, Sriperumbudur, et al. (2012) analyzed the problem of choosing a kernel for $\widehat{\textsc{mmd}}_s^2$. Recall from (6.2) that

$$\frac{1}{V_s^{(m)}}\left(\widehat{\textsc{mmd}}_s^2 - \textsc{mmd}^2\right) \xrightarrow{D} \mathcal{N}(0, 1),$$

using the variance $V_s^{(m)}$ from (6.3), which is the same under the null and the alternative. Thus, the asymptotic test threshold for the streaming estimator is simply

$$c_\alpha := \sqrt{V_s^{(m)}}\,\Phi^{-1}(1-\alpha),$$

where Φ is the CDF of a standard normal random variable. Since $V_s^{(m)}$ is unknown in practice, we instead use the estimator of (6.4):

$$\hat{c}_\alpha := \sqrt{\widehat{V}_s^{(m)}}\,\Phi^{-1}(1-\alpha).$$

The asympotic power of such a test is, using $\Pr_{H_1}$ to denote probability under the alternative H_1,

$$\begin{aligned}
\Pr_{H_1}\left(\widehat{\textsc{mmd}}_s^2 > \hat{c}_\alpha\right) &= \Pr_{H_1}\left(\frac{\widehat{\textsc{mmd}}_s^2 - \textsc{mmd}^2}{\sqrt{V_s^{(m)}}} > \frac{\hat{c}_\alpha - \textsc{mmd}^2}{\sqrt{V_s^{(m)}}}\right) \\
&= \Pr_{H_1}\left(\frac{\widehat{\textsc{mmd}}_s^2 - \textsc{mmd}^2}{\sqrt{V_s^{(m)}}} > \sqrt{\frac{\widehat{V}_s^{(m)}}{V_s^{(m)}}}\,\Phi^{-1}(1-\alpha) - \frac{\textsc{mmd}^2}{\sqrt{V_s^{(m)}}}\right) \\
&\to 1 - \Phi\left(\Phi^{-1}(1-\alpha) - \frac{\textsc{mmd}^2}{\sqrt{V_s^{(m)}}}\right).
\end{aligned}$$

The power is thus asymptotically maximized when $\mathrm{MMD}^2/\sqrt{V_s^{(m)}}$ is maximal. In practice, we optimize $\widehat{\mathrm{MMD}}_s^2/\sqrt{\widehat{V}_s^{(m)}}$.

We will call this quantity $t_s := \mathrm{MMD}^2/\sqrt{V_s^{(m)}}$, and its estimator $\hat{t}_s := \widehat{\mathrm{MMD}}_s^2/\sqrt{\widehat{V}_s^{(m)}}$, as it follows the form of a t-statistic for MMD_s^2.

Gretton, Sriperumbudur, et al. (2012, Theorem 1) proved that, when considering nonnegative combinations of a fixed set of base kernels, the maximum of the ratio estimate approaches the maximum of the ratio at a rate $O_P\left(m^{-\frac{1}{3}}\right)$, and that the kernel achieving the maximum ratio estimate converges in probability to the kernel achieving the maximum ratio.

6.3.8 Pairwise t-statistic

We can in fact make a similar argument for $\widehat{\mathrm{MMD}}_U^2$.

Under H_0, $m\,\widehat{\mathrm{MMD}}_U^2$ converges in distribution to an infinite mixture of χ^2 random variables, with weights depending on the (unknown) distributions P and Q as well as k (Gretton, Borgwardt, et al. 2012); c_α is thus difficult to evaluate in closed form. We can, however, estimate a data-dependent threshold $\hat{c}_\alpha$ according to a permutation test: randomly partition the data points $X \cup Y$ into X' and Y' many times, evaluate $\widehat{\mathrm{MMD}}_U^2(X', Y')$ to approximate the null distribution, and then estimate the $(1-\alpha)$th quantile c_α from these samples.[2]

Under the alternative H_1, however, recall from (6.5) that the distribution is asymptotically normal:

$$\frac{\widehat{\mathrm{MMD}}_U^2 - \mathrm{MMD}^2}{\sqrt{V_U^{(m)}}} \xrightarrow{D} \mathcal{N}(0,1)$$

using $V_U^{(m)}$ from (6.6). We can thus compute the test power as:

$$\begin{aligned}\Pr_{H_1}\left(m\,\widehat{\mathrm{MMD}}_U^2 > \hat{c}_\alpha\right) &= \Pr_{H_1}\left(\frac{\widehat{\mathrm{MMD}}_U^2 - \mathrm{MMD}^2}{\sqrt{V_U^{(m)}}} > \frac{\hat{c}_\alpha}{m\sqrt{V_U^{(m)}}} - \frac{\mathrm{MMD}^2}{\sqrt{V_U^{(m)}}}\right)\\ &\to 1 - \Phi\left(\frac{c_\alpha}{m\sqrt{V_U^{(m)}}} - \frac{\mathrm{MMD}^2}{\sqrt{V_U^{(m)}}}\right).\end{aligned}$$

Defining

$$\tau_U := \frac{\mathrm{MMD}^2}{\sqrt{V_U^{(m)}}} - \frac{c_\alpha}{m\sqrt{V_U^{(m)}}} \qquad \text{and} \qquad \hat{\tau}_U := \frac{\widehat{\mathrm{MMD}}_U^2}{\sqrt{\widehat{V}_U^{(m)}}} - \frac{c_\alpha}{m\sqrt{\widehat{V}_U^{(m)}}},$$

[2] Gretton, Fukumizu, et al. (2009) proposed a way to estimate c_α without permutation tests by examining the eigenvalues of the data Gram matrix, but a recent cache-efficent implementation of permutation tests in the Shogun toolbox (Sonnenburg et al. 2010) is actually significantly quicker to compute than this estimate. We thus only consider permutation tests here.

we see that the power is maximal when τ_U is maximal. In practice, we maximize its estimator $\hat{\tau}_U$.

But note that c_α and MMD are constant as the sample size m increases, and $V_U^{(m)}$ is $O\left(\frac{1}{m}\right)$. For large m, therefore, the first term dominates the second, and it suffices to maximize just the first term

$$t_U := \frac{\text{MMD}^2}{\sqrt{V_U^{(m)}}} \qquad \text{or its estimator} \qquad \hat{t}_U := \frac{\widehat{\text{MMD}}_U^2}{\sqrt{\widehat{V}_U^{(m)}}},$$

which (appealingly) is of the same form as the t-statistic in the streaming setting. When the test is on the cusp of rejection, however, $\hat{c}_\alpha \approx m\widehat{\text{MMD}}^2$, and thus the two terms are of similar magnitude; additionally, using t_U can lead to asymptotic power predictions no smaller than $\frac{1}{2}$. We will see in the experiments section that for simple problems, while t_U gives inaccurate estimates of the asymptotic power but τ_U's are reasonable, the maximum of $\hat{t}_U$ often coincides with that of $\hat{\tau}_U$.

Gradients As mentioned previously, complex kernel functions are far more powerful in some domains than simple families such as Gaussian RBFs. We would like to be able to choose kernels by, for example, passing inputs through a deep network to learn a representation, and then comparing those learned representations with a standard kernel. It is far easier to optimize over such complex kernel classes, however, when gradient information is available. It is thus important to note that $\hat{t}_U$ is differentiable in k: $\widehat{\text{MMD}}_U^2$ is an average of applications of k, and $\widehat{V}_U^{(m)}$ (6.7) is based on terms of a similar form.[3]

In fact, we can also obtain stochastic gradients of $\hat{c}_\alpha$ with respect to k. Let $\Pi = \{\pi_1, \dots, \pi_N\}$ denote the set of permutations applied to the data, and X'_π, Y'_π the result of applying one of those permutations, so that our approximate sample from the null distribution is $\{\eta_{\pi_i} := \widehat{\text{MMD}}_U^2(X'_{\pi_i}, Y'_{\pi_i})\}_{i=1}^N$. Let J be the nearest integer to $(1-\alpha)N$, and j be the index of the permutation achieving the Jth largest η_π value. Then $\hat{c}_\alpha^{(\Pi)} = \eta_{\pi_j}$ is the test threshold corresponding to the set of permutations Π. The gradient of $\hat{c}_\alpha^{(\Pi)}$ by k is simply the gradient by k of η_{π_j}. But, assuming that $\widehat{\text{MMD}}_U$ is Lipschitz in the parameterization of k, the Leibniz rule tells us that

$$\mathbb{E}_\Pi\left[\nabla_k \hat{c}_\alpha^{(\Pi)}\right] = \nabla_k\left[\mathbb{E}_\Pi\, \hat{c}_\alpha^{(\Pi)}\right] = \nabla_k \hat{c}_\alpha.$$

6.4 Experiments

We will now study the effectiveness of maximizing $\hat{\tau}_U$ and $\hat{t}_U$ versus that of maximizing the MMD on synthetic problems. Further experiments on more realistic problems are left to future work.

We do simple bandwidth selection for Gaussian RBF kernels. For each pair of distributions, we draw 100 samples (X, Y) and compute the criteria and run a permutation test for each of 30 logarithmically-spaced values for σ from $10^{-1.7}$ to $10^{1.7}$. We use 1 000 permutations in the tests, which are implemented in the `feature/bigtest` branch of Shogun (Sonnenburg et al. 2010).

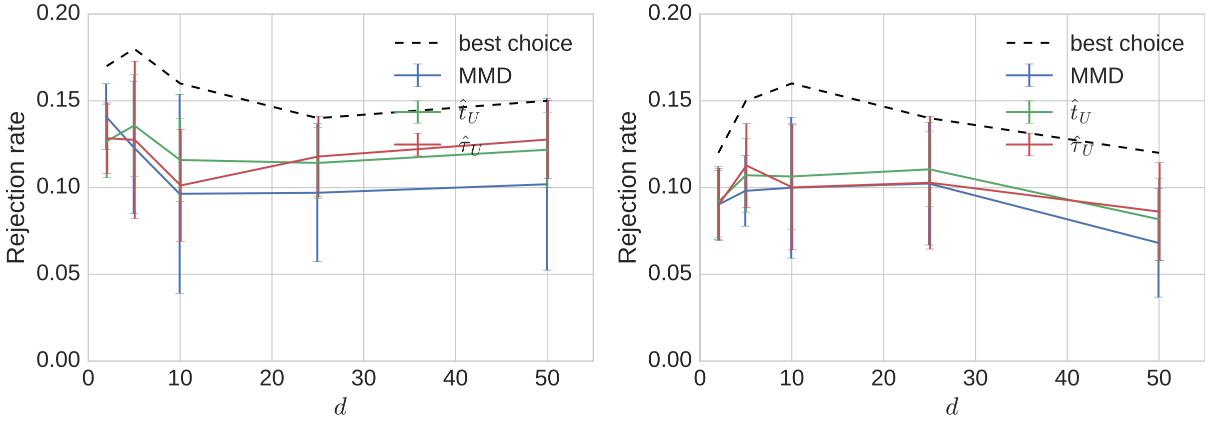
[3] The gradient is quite long to write out, but it is amenable to automatic differentiation e.g. in Theano (The Theano Development Team et al. 2016).

All tests use an allowed false positive rate of 10%. In the results, "best choice" refers to the bandwidth with the maximal empirical power.

6.4.1 Same Gaussian

In this situation, the null distribution holds: $P = Q = \mathcal{N}(0, I)$. Figure 6.1 verifies that the stated false positive level is adhered to by each proposed method.

Note that "best choice" here gives a test slightly larger than desired, because it is chosen to maximize the rejection rate on the same datasets as it is plotted on.



(a) $m = 100$

(b) $m = 500$

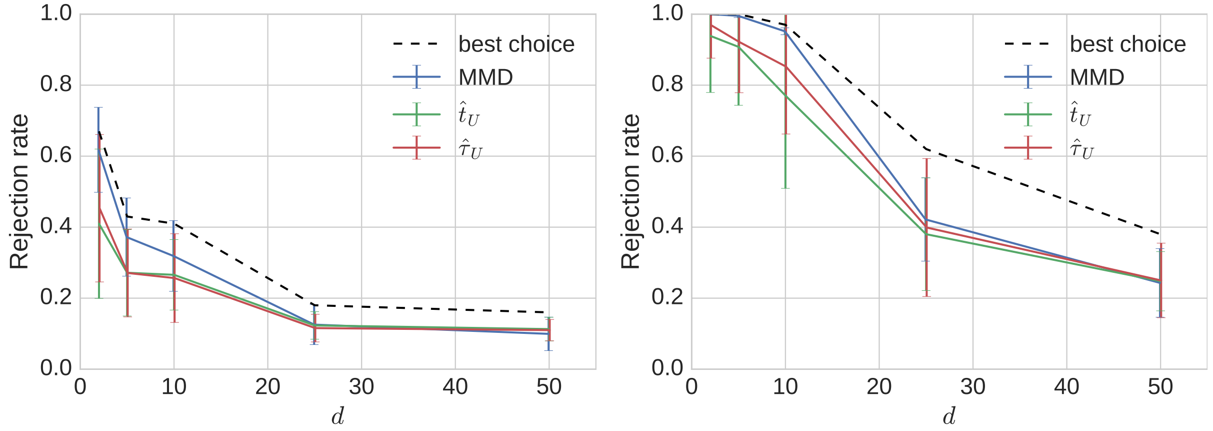
Figure 6.1: Same Gaussian problem: mean and standard deviation of test powers for increasing dimension.

6.4.2 Gaussian variance difference

Here we test $P = \mathcal{N}(0, I)$ versus $Q = \mathcal{N}(0, I + e_1 e_1^\mathsf{T})$. (In Q, the variance of the first dimension is twice that in the other dimensions.) Figure 6.2 shows results; maximizing the MMD actually slightly outperforms maximizing $\hat{t}_U$ or $\hat{\tau}_U$.

Figure 6.3 breaks down the difference in the case $d = 2$, $m = 100$. We can see that maximizing the MMD usually picked banwdiths near the peak power, whereas $\hat{t}_U$ and $\hat{\tau}_U$ often picked bandwidths either somewhat larger than the peak or occasionally much smaller. Figure 6.4 shows the criteria used to select those bandwidths, including their asymptotic values based on the true MMD and the asymptotic variance of the $\widehat{\text{MMD}}_U^2$ estimator of normal distributions. (For τ_U, we used the mean value of the permutation-based $\hat{c}_\alpha$ across repeated draws from the dataset for the asymptotic value of c_α.)

We can see here that the difference in performance is not just a poor variance estimate, but that the asymptotic values of τ_U and especially t_U are less suited to bandwidth selection here than simply maximizing the MMD. Given the lack of theory about the power of tests based on maximizing the MMD, this difference is somewhat difficult to explain further.



(a) $m = 100$

(b) $m = 500$

Figure 6.2: Gaussian variance difference problem: mean and standard deviation of test powers for increasing dimension.

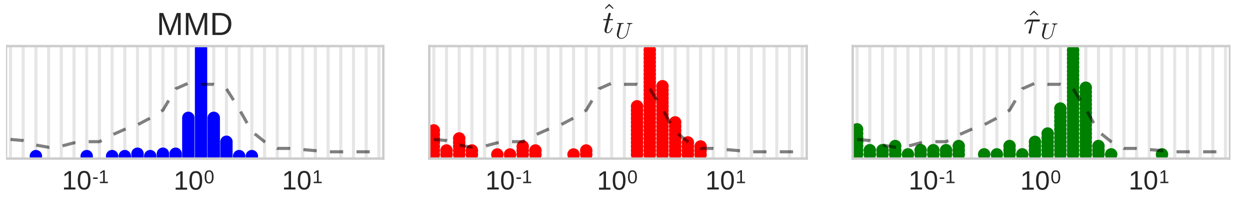
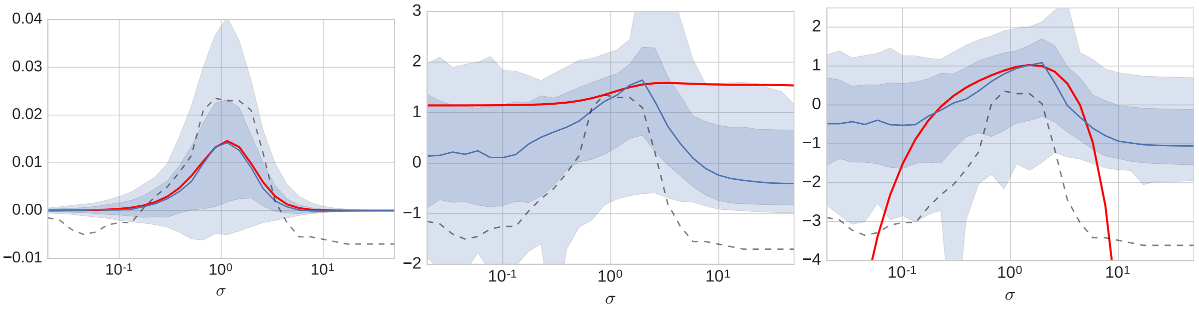


Figure 6.3: Chosen bandwidths for the three methods for Gaussian variance difference for $d = 2$, $m = 100$. Vertical gray lines represent the candidate bandwidths, in log scale; bars show the number of times each bandwidth was chosen. The gray dashed line shows the empirical power of each bandwidth, so that e.g. the central bandwidth 1 achieved power about 0.7.



(a) MMD_U^2

(b) t_U

(c) τ_U

Figure 6.4: The various critera for the Gaussian variance difference problem. In each figure, the blue line shows the median of the estimator, darker blue region 68% scatter, and lighter blue region 95% scatter; thick red lines show the asymptotic value of the quantity in question. On a separate vertical scale (not labeled), gray dashed lines show the empirical power of each bandwidth, so that e.g. the central bandwidth 1 achieved power about 0.7.

6.4.3 Blobs

We now consider the blobs problem of Gretton, Sriperumbudur, et al. (2012): P is a 5×5 grid of two-dimensional standard normal components, with spacing 10 between the centers. Q is laid out identically, but each mixture component is $\mathcal{N}\left(\mu, \begin{bmatrix} 1 & \frac{\varepsilon-1}{\varepsilon+1} \\ \frac{\varepsilon-1}{\varepsilon+1} & 1 \end{bmatrix}\right)$, so that the ratio of eigenvalues in its variance is ε. Note that at $\varepsilon = 1$, $P = Q$. An example grid is shown in Figure 6.5.

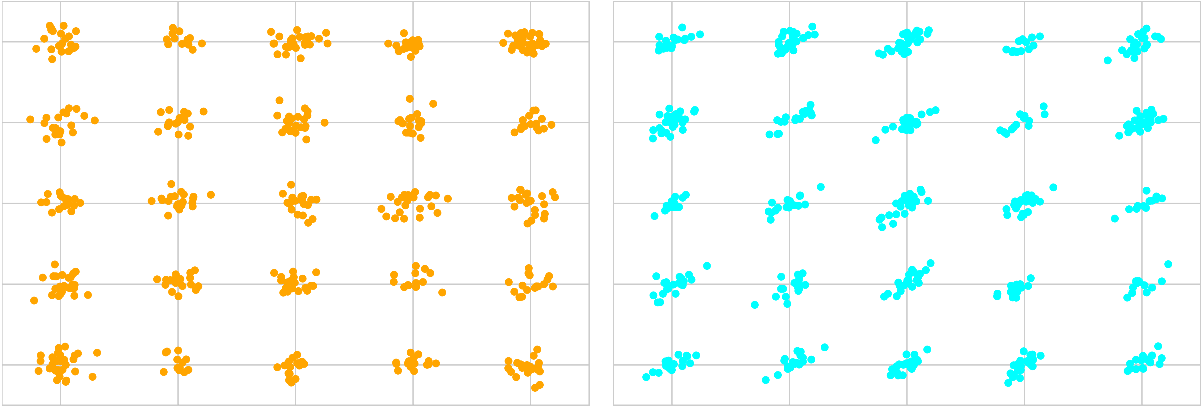
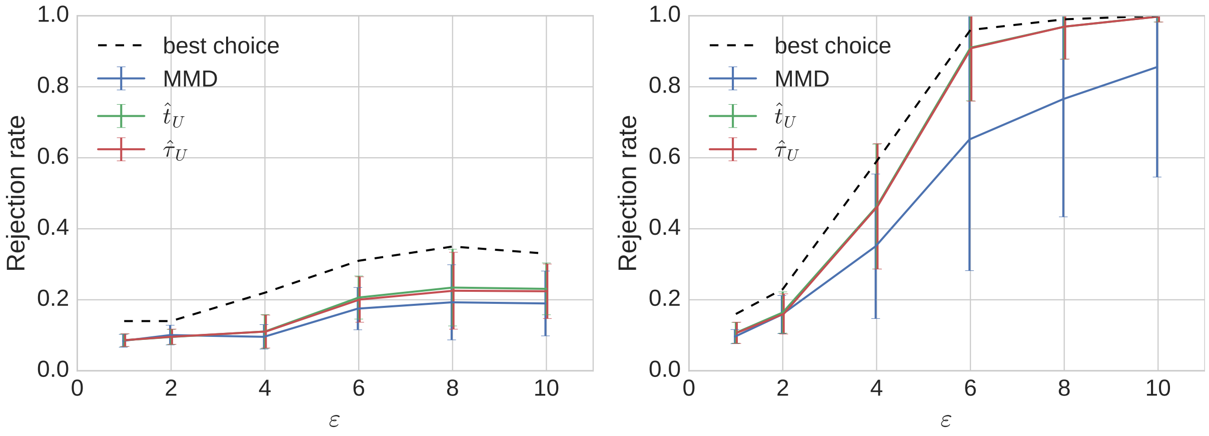


Figure 6.5: A sample from the Blobs problem, with $m = 500$, $\varepsilon = 6$.

Figure 6.6 shows results; here, $\hat{t}_U$ and $\hat{\tau}_U$ each outperform MMD, especially when $m = 500$, and are nearly optimal.

We again take a closer look at the criteria, here where $\varepsilon = 6, m = 500$. Figure 6.7 shows the selected bandwidths; we can see that in this case, maximizing the MMD usually either picked bandwidths slightly too large or sometimes much too large, whereas $\hat{t}_U$ and $\hat{\tau}_U$ both consistently selected bandwidths around the peak power.

Figure 6.8 shows the criteria used to select those bandwidths. Here, although asymptotic values of the variance are available, 500 samples (on expectation, only 20 per blob) is not enough for it to converge well to its asymptotic value. Thus we use the empirical variance of the MMD estimator across our repeated dataset samples instead. We can see that in this case, although the true MMD peak is not too bad (it is only a little large), for large bandwidths the MMD estimator has a very high variance, and thus maximizing the MMD often picks a very largue bandwidth value. t_U and τ_U, by contrast, both asymptotically peak in the correct location and their estimates do not vary too widely other than in the cases where the MMD blows up, in which case its variance increases even more and so an already-bad location only seems worse than it really is.



(a) m = 100

(b) m = 500

Figure 6.6: Blobs problem: mean and standard deviation of test powers for increasing eigenvalue ratio.

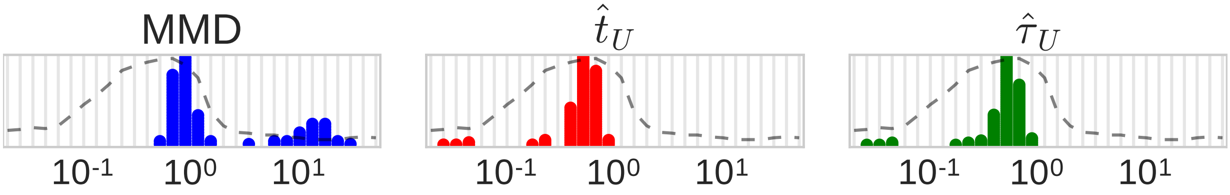
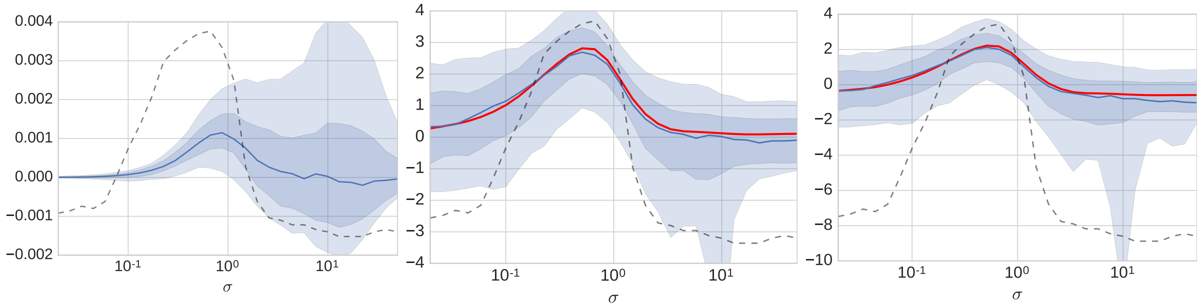


Figure 6.7: Chosen bandwidths for the three methods for the Blobs problem with $\varepsilon = 6$, $m = 500$. Figures as in Figure 6.3.



(a) MMD_U^2

(b) t_U

(c) τ_U

Figure 6.8: The various critera for the blobs problem. Figures as in Figure 6.4, except that red lines use empirical variance across the samples rather than asymptotics.

Chapter 7

Active search for patterns

We will now change focus slightly, and consider another problem setting in which collections of data play a key role.

Consider a function containing interesting patterns that are defined only over a region of space. For example, if you view the direction of wind as a function of geographical location, it defines fronts, vortices, and other weather patterns, but those patterns are defined only in the aggregate. If we can only measure the direction and strength of the wind at point locations, we then need to infer the presence of patterns over broader spatial regions.

Many other real applications also share this feature. For example, an autonomous environmental monitoring vehicle with limited onboard sensors needs to strategically plan routes around an area to detect harmful plume patterns on a global scale (Valada et al. 2012). In astronomy, projects like the Sloan Digital Sky Survey (Eisenstein et al. 2011) search the sky for large-scale objects such as galaxy clusters. Biologists investigating rare species of animals must find the ranges where they are located and their migration patterns (Brown et al. 2014). We aim to use active learning to search for such global patterns using as few local measurements as possible.

This bears some resemblance to the artistic technique known as pointillism, where the painter creates small and distinct dots each of a single color, but when viewed as a whole they reveal a scene. Pointillist paintings typically use a denser covering of the canvas, but in our setting, "observing a dot" is expensive. Where should we make these observations in order to uncover interesting regions as quickly as possible?

We propose a probabilistic solution to this problem, known as *active pointillistic pattern search* (APPS). We assume we are given a predefined list of candidate regions and a classifier that estimates the probability that a given region fits the desired pattern. Our goal is then to find as many regions that are highly likely to match the pattern as we can. We accomplish this by sequentially selecting point locations to observe so as to approximately maximize expected reward.

7.1 Related work

Our concept of active pattern search falls under the broad category of *active learning* (Settles 2012), where we seek to sequentially build a training set to achieve some goal as fast as possible.

Our focus solely on finding positive ("interesting") regions, rather than attempting to learn to discriminate accurately between positives and negatives, is similar to the problem previously described as *active search* (Garnett et al. 2012). In previous work on active search, however, it has been assumed that the labels of interest can be revealed directly. In active pattern search, on the other hand, the labels are never revealed but must be inferred via a provided classifier. This indirection increases the difficulty of the search task considerably.

In *Bayesian optimization* (Osborne et al. 2009; Brochu et al. 2010), we seek to find the global optimum of an expensive black-box function. Bayesian optimization provides a model-based approach where a Gaussian process (GP) prior is placed on the objective function, from which a simpler acquisition function is derived and optimized to drive the selection procedure. Tesch et al. (2013) extend this idea to optimizing a latent function from binary observations. Our proposed active pattern search also uses a Gaussian process prior to model the unknown underlying function and derives an acquisition function from it, but differs in that we seek to identify entire *regions* of interest, rather than finding a single optimal value.

Another intimately related problem setup is that of *multi-arm bandits* (Auer et al. 2002), with more focus on analysis of the cumulative reward over all function evaluations. Originally, the goal was to maximize the expectation of a random function on a discrete set; a variant considers the optimization in continuous domains (Kroemer et al. 2010; Niranjan et al. 2010). However, like Bayesian optimization, multi-arm bandit problems usually do not consider discriminating a regional pattern.

Level set estimation (Low et al. 2012; Gotovos et al. 2013), rather than finding optima of a function, seeks to select observations so as to best discriminate the portions of a function above and below a given threshold. This goal, though related to ours, aims to directly map a portion of the function on the input space rather than seeking out instances of patterns. LSE algorithms can be used to attempt to find some simple types of patterns, e.g. areas with high mean.

APPS can be viewed as a generalization of *active area search* (AAS) (Y. Ma, Garnett, et al. 2014), which is a considerably simpler version of active search for region-based labels. In AAS, the label of a region is only determined by whether its mean value exceeds some threshold. APPS allows for arbitrary classifiers rather than simple thresholds, and in some cases its expected reward can still be computed analytically. This extends the usefulness of this class of algorithms considerably.

7.2 Problem formulation

There are three key components of the APPS framework: a function f which maps input covariates to data observations, a predetermined set of regions wherein instances of function patterns are expected, and a classifier that evaluates the salience of the pattern of function values in each region. We define $f\colon \mathbb{R}^m \to \mathbb{R}$ to be the function of interest,[1] which can be observed at any location $x \in \mathbb{R}^m$ to reveal a noisy observation z. We assume the observation model $z = f(x) + \varepsilon$, where $\varepsilon \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$. We suppose that a set of regions where matching patterns might be found is

[1] For clarity, in this and the next sections we will focus on scalar-valued functions f. The extension to vector-valued functions is straightforward; we consider such a case in the experiments.

predefined, and will denote these $\{g_1, \ldots, g_k\}$; $g_i \subset \mathbb{R}^m$. Finally, for each region g, we assume a classifier h_g which evaluates f on g and returns the probability that it matches the target pattern, which we call *salience*: $h_g(f) = h(f; g) \in [0, 1]$, where the mathematical interpretation of h_g is similar to a functional of f. Classifier forms are typically the same for all regions with different parameters.

Unfortunately, in general, we will have little knowledge about f other than the limited observations made at our selected set of points. Classifiers which take functional inputs (such as our assumed h_g) generally do not account for uncertainty in their inputs, which should be inversely related to the number of observed data points. We thus consider the probability that $h_g(f)$ is high enough, marginalized across the range of functions f that might match our observations. As is common in nonparametric Bayesian modeling, we model f with a Gaussian process (GP) prior; we assume that hyperparameters, including prior mean and covariance functions, are set by domain experts. Given a dataset $\mathcal{D} = (X, z)$, we define

$$f \sim \mathcal{GP}(\mu, \kappa); \qquad f \mid \mathcal{D} \sim \mathcal{GP}(\mu_{f|\mathcal{D}}, \kappa_{f|\mathcal{D}}),$$

to be a given GP prior and its posterior conditioned on $\mathcal{D}$, respectively. Thus, since f is a random variable, we can obtain the marginal probability that g is salient,

$$T_g(\mathcal{D}) = \mathbb{E}_f\big[h_g(f) \mid \mathcal{D}\big]. \tag{7.1}$$

We then define a matching region as one whose marginal probability passes a given threshold θ. Unit reward is assigned to each matching region g:

$$r_g(\mathcal{D}) \coloneqq \mathbb{1}\big\{T_g(\mathcal{D}) > \theta\big\}.$$

We make two assumptions regarding the interactive procedure. The first is that once a region is flagged as potentially matching (i.e., its marginal probability exceeds θ), it will be immediately flagged for further review and no longer considered during the run. The second is that the data resulting from this investigation will not be made immediately available during the course of the algorithm; rather the classifiers h_g will be trained offline. We consider both of these assumptions to be reasonable when the cost of investigation is relatively high and the investigation collects different types of data. For example, if the algorithm is being used to run autonomous sensors and scientists collect separate data to follow up on a matching region, these assumptions allow the autonomous sensors to continue in parallel with the human intervention, and avoid the substantial complexity of incorporating a completely different modality of data into the modeling process.

Garnett et al. (2012) attempt to maximize their reward at the end of a fixed number of queries. Directly optimizing that goal involves an exponential lookahead process. However, this can be approximated by a greedy search like the one we perform. Similarly, one could attempt to maximize the area under the recall curve through the search process. This also requires an intractable amount of computation which is often replaced with a greedy search.

We now write down the greedy criterion our algorithm seeks to optimize. Define $\mathcal{D}_t$ to be the already collected (noisy) observations of f before time step t and $\mathcal{G}_t = \{g : T_g(\mathcal{D}_\tau) \leq \theta, \forall \tau \leq t\}$ to be the set of remaining search subjects, those regions which are not yet confidently salient; we aim to greedily maximize the sum of rewards over all the regions in $\mathcal{G}_t$ in expectation,

$$\max_{x_*} \mathbb{E}\left[\sum_{g\in\mathcal{G}_t} r_g(\mathcal{D}_*) \,\middle|\, x_*, \mathcal{D}_t\right], \tag{7.2}$$

where $\mathcal{D}_*$ is the (random) dataset augmented with x_*.

This criterion satisfies a desirable property: when the regions are uncoupled and the classifier h_g is probit-linear, the point that maximizes (7.2) in each region also minimizes the variance of that region's label (Section 7.3.2).

7.3 Method

For the aim of maximizing the greedy expected reward of finding matching patterns (7.2), a more careful examination of the GP model can yield a straightforward sampling method. This method, in the following, turns out to be quite useful in APPS problems with rather complex classifiers. Section 7.3.1 introduces an analytical solution in an important special case.

At each step, given $\mathcal{D}_t = (X, z)$ as the set of any already collected (noisy) observations of f and x_* as any potential input location, we can assume the distribution of possible observations $z_* = f(x_*) + \varepsilon$ as

$$z_* \mid x_*, \mathcal{D}_t \ \sim\ \mathcal{N}\big(\mu_{f|\mathcal{D}_t}(x_*),\ \kappa_{f|\mathcal{D}_t}(x_*, x_*) + \sigma^2\big). \tag{7.3}$$

Conditioned on an observation value z_*, we can update our GP model to include the new observation (x_*, z_*), which further affects the marginal distribution of region classifier outputs and thus the probability this region is matching. With $\mathcal{D}_* = \mathcal{D}_t \cup \big\{(x_*, z_*)\big\}$ as the updated dataset, we use $r_g(\mathcal{D}_*)$ to be the updated reward of region g. The utility of this proposed location x_* for region g is thus measured by the *expected* reward function, marginalizing out the unknown observation value z_*:

$$\begin{aligned} u_g(x_*, \mathcal{D}_t) &\coloneqq \mathbb{E}_{z_*}\big[r_g(\mathcal{D}_*) \mid x_*, \mathcal{D}_t\big] && (7.4)\\ &= \Pr\big\{T_g(\mathcal{D}_*) > \theta \mid x_*, \mathcal{D}_t\big\}. && (7.5)\end{aligned}$$

Finally, in active pointillistic pattern search, we select the next observation location x_* by considering its expected reward over the remaining regions:

$$x_* = \operatorname*{argmax}_x\ u(x, \mathcal{D}_t) = \operatorname*{argmax}_x \sum_{g\in\mathcal{G}_t} u_g(x, \mathcal{D}_t). \tag{7.6}$$

For the most general definition of the region classifier h_g, the basic algorithm is to compute (7.4) and thus (7.6) via sampling at two stages:

1. Sample the outer variable z_* in (7.4) according to (7.3).
2. For every draw of z_*, sample enough of $(f \mid \mathcal{D}_*)$ to compute the marginal reward $T_g(\mathcal{D}_*)$ in (7.1), in order to obtain one draw for the expectation in (7.4).

To speed up the process, we can evaluate (7.6) for a subset of possible x_* values, as long as a good action is likely to be contained in the set.

7.3.1 Analytic expected utility for functional probit models

For a broad family of classifiers, those formed by a probit link function of any affine functional of f (7.7), we can compute both (7.1) and (7.5) analytically. Thus, we can efficiently perform exact searches for potentially complex patterns defined by probit-linear classifiers.

Suppose we have observed data $\mathcal{D}$, yielding the posterior $p(f \mid \mathcal{D}) = \mathcal{GP}(f; \mu_{f|\mathcal{D}}, \kappa_{f|\mathcal{D}})$. Let L_g be a linear functional, $L_g \colon f \mapsto L_g f \in \mathbb{R}$, associated with region g. The family of classifiers is:

$$h_g(f) = \Phi(L_g f + b_g), \tag{7.7}$$

where Φ is the cumulative distribution function of the standard normal and $b \in \mathbb{R}$ is an offset. Two examples of such functionals are:

- $L_g f \colon f \mapsto \frac{c}{|g|} \int_g f(x)\,\mathrm{d}x$, where $|g|$ is the volume of region $g \subset \mathbb{R}^m$. Here $L_g f$ is the mean value of f on g, scaled by an arbitrary $c \in \mathbb{R}$. When $|c| \to \infty$ the model becomes quite similar to that of Y. Ma, Garnett, et al. (2014).
- $L_g f \colon f \mapsto w^\top f(\Xi)$, where Ξ is a finite set of fixed points $\{\xi_i\}_{i=1}^{|\Xi|}$, and $w \in \mathbb{R}^{|\Xi|}$ is an arbitrary vector. This mapping applies a linear classifier to a fixed, discrete set of values from f.

As Gaussian processes are closed under linear transformations, $Lf + b$ has a normal distribution:

$$Lf + b \sim \mathcal{N}(L\mu_{f|\mathcal{D}} + b,\, L^2\kappa_{f|\mathcal{D}}),$$

where L^2 is the bilinear form defined by $L^2\kappa := L\big[L\kappa(x,\cdot)\big] = L\big[L\kappa(\cdot,x')\big]$. For the specific cases above, we can explicitly calculate the mean and variance of $Lf + b$: for $Lf = w^\top f(\Xi)$

$$\mathbb{E}_f[Lf \mid \mathcal{D}] = w^\top \mu_{f|\mathcal{D}}(\Xi) \qquad \operatorname{Var}_f[Lf \mid \mathcal{D}] = w^\top \kappa_{f|\mathcal{D}}(\Xi,\Xi)\, w$$

and for $Lf = \frac{c}{|g|}\int_g f(x)\,\mathrm{d}x$

$$\mathbb{E}_f[Lf \mid \mathcal{D}] = \frac{c}{|g|}\int_g \mu_{f|\mathcal{D}}(x)\,\mathrm{d}x \qquad \operatorname{Var}_f[Lf \mid \mathcal{D}] = \frac{c^2}{|g|^2}\iint_{g^2} \kappa_{f|\mathcal{D}}(x,x')\,\mathrm{d}x\,\mathrm{d}x'.$$

For certain classes of covariance functions κ, the above integrals are tractable; they occur when estimating integrals via *Bayesian quadrature*, also known as *Bayesian Monte Carlo* (Rasmussen and Ghahramani 2003).

Then we have the marginal probability that g is salient (7.1) in closed form:

$$\begin{aligned} T_g(\mathcal{D}) &= \mathbb{E}_f\big[h_g(f) \mid \mathcal{D}\big] \\ &= \mathbb{E}_f\big[\Phi(Lf + b) \mid \mathcal{D}\big] \\ &= \Phi\left(\frac{L\mu_{f|\mathcal{D}} + b}{\sqrt{1 + L^2\kappa_{f|\mathcal{D}}}}\right), \end{aligned}$$

using the fact that if $A \sim \mathcal{N}(\mu, \sigma^2)$, then $\mathbb{E}[\Phi(A)] = \Phi\left(\mu/\sqrt{1+\sigma^2}\right)$.

Now we turn to the expected utility of a new observation (7.5). Consider a potential observation location x_*, and again define $\mathcal{D}_* := \mathcal{D} \cup \{(x_*, z_*)\}$. Then $u_g(x_*, \mathcal{D})$ is:

$$\begin{aligned} u_g(x_*, \mathcal{D}) &= \Pr\left[\Phi\left(\frac{L\mu_{f\mid\mathcal{D}_*} + b}{\sqrt{1 + L^2\kappa_{f\mid\mathcal{D}_*}}}\right) > \theta \;\middle|\; x_*, \mathcal{D}\right] \\ &= \Pr\left[\frac{L\mu_{f\mid\mathcal{D}_*} + b}{\sqrt{1 + L^2\kappa_{f\mid\mathcal{D}_*}}} > \Phi^{-1}(\theta) \;\middle|\; x_*, \mathcal{D}\right], \end{aligned} \tag{7.8}$$

where Φ^{-1} is the inverse of the normal CDF.

Letting the variance of the new point x_* given the dataset $\mathcal{D}$ be denoted by

$$V_{*\mid\mathcal{D}} := \operatorname{Var}[z_* \mid \mathcal{D}] = \kappa_{f\mid\mathcal{D}}(x_*, x_*) + \sigma^2,$$

we have

$$\begin{aligned} L^2\kappa_{f\mid\mathcal{D}_*} &= L^2\left[\kappa_{f\mid\mathcal{D}}(x, x') - \kappa_{f\mid\mathcal{D}}(x, x_*)\, V_{*\mid\mathcal{D}}^{-1}\, \kappa_{f\mid\mathcal{D}}(x_*, x')\right] \\ &= L^2\kappa_{f\mid\mathcal{D}} - L\left[\kappa_{f\mid\mathcal{D}}(\cdot, x_*)\right]\, V_{*\mid\mathcal{D}}^{-1}\, L\left[\kappa_{f\mid\mathcal{D}}(x_*, \cdot)\right], \end{aligned} \tag{7.9}$$

which does not depend on z_*.

Next, consider the distribution of $L\mu_{f\mid\mathcal{D}_*}$. If we knew the observation value z_*, we could compute the updated posterior mean as

$$\mu_{f\mid\mathcal{D}_*}(x) = \mu_{f\mid\mathcal{D}}(x) + \kappa_{f\mid\mathcal{D}}(x, x_*)\, V_{*\mid\mathcal{D}}^{-1}\left(z_* - \mu_{f\mid\mathcal{D}}(x_*)\right).$$

But, thanks to the linearity of L and the known Gaussian distribution on z_*, the updated posterior mean of Lf is also normally distributed with

$$L\mu_{f\mid\mathcal{D}_*} \mid x_*, \mathcal{D} \sim \mathcal{N}\left(L\mu_{f\mid\mathcal{D}}, V_{*\mid\mathcal{D}}^{-1}\, L\left[\kappa_{f\mid\mathcal{D}}(\cdot, x_*)\right]^2\right) \tag{7.10}$$

and so, using (7.10) in (7.8), we can finally compute the desired expected reward $u_g(x_*, \mathcal{D})$ in closed form:

$$u_g(x_*, \mathcal{D}) = \Phi\left(\frac{L_g\mu_{f\mid\mathcal{D}} + b - \sqrt{1 + L_g^2\kappa_{f\mid\mathcal{D}_*}}\,\Phi^{-1}(\theta)}{\sqrt{V_{*\mid\mathcal{D}}^{-1} L_g\left[\kappa_{f\mid\mathcal{D}}(\cdot, x_*)\right]^2}}\right). \tag{7.11}$$

7.3.2 Analysis for independent regions

The analytical solution to (7.5) by (7.11) enables us to further study the theory behind the exploration/exploitation tradeoff of APPS in one nontrivial case: when all regions are approximately independent. This assumption allows us to ignore the effect a data point has on regions other than its own. We will answer two questions in this case: which region will APPS explore next, and what location will be queried for that region.

Define

$$\rho_g(x_*)^2 := \frac{V_{*|\mathcal{D}}^{-1} L_g \left[\kappa_{f|\mathcal{D}}(\cdot, x_*)\right]^2}{1 + L_g^2 \kappa_{f|\mathcal{D}}} = \frac{\operatorname{Var}\left[L_g \mu_{f|\mathcal{D}_*} \mid x_*, \mathcal{D}\right]}{1 + \operatorname{Var}\left[L_g f + b \mid \mathcal{D}\right]}, \tag{7.12}$$

which in some sense denotes how informative the observation z_* is expected to be to the label of its region g. With this notation, (7.9) becomes

$$1 + L_g^2 \kappa_{f|\mathcal{D}_*} = (1 - \rho_g(x_*)^2)(1 + L_g^2 \kappa_{f|\mathcal{D}}).$$

Assume for now that $\theta > 0.5$. (Our conclusions remain true for any θ, but for simplicity we consider only the common case here.) Then we can define how close g is to receiving a reward by

$$R_g := \frac{\Phi^{-1}(T_g(\mathcal{D}))}{\Phi^{-1}(\theta)} = \frac{L_g \mu_{f|\mathcal{D}} + b}{\Phi^{-1}(\theta)\sqrt{1 + L_g^2 \kappa_{f|\mathcal{D}}}}. \tag{7.13}$$

Thus the utility (7.11) becomes, using (7.12) and (7.13):

$$u_g(x_*, \mathcal{D}) = \Phi\left(\Phi^{-1}(\theta) \frac{R_g - \sqrt{1 - \rho_g(x_*)^2}}{\rho_g(x_*)}\right).$$

We can now see by taking partial derivatives that for any region not currently carrying a reward:

1. For any region g, $u_g(x, \mathcal{D})$ is maximized by choosing an x that yields $\rho_g^* := \max_x \rho_g(x)$.
2. If two regions g and g' can be equally explored ($\rho_g^* = \rho_{g'}^*$), then the region with higher probability of matching (higher R) will be selected.
3. If two regions are equally likely to match the desired pattern ($R_g = R_{g'}$), the more explorable region (that with a larger ρ^*) will be selected.
4. In general, APPS will trade off the two factors by maximizing $\left(R_g - \sqrt{1 - (\rho_g^*)^2}\right) / \rho_g^*$.

7.4 Empirical evaluation

We now turn to an empirical evaluation of our framework, in three different settings and with three different classifiers. Code and data for these experiments is available online.[2]

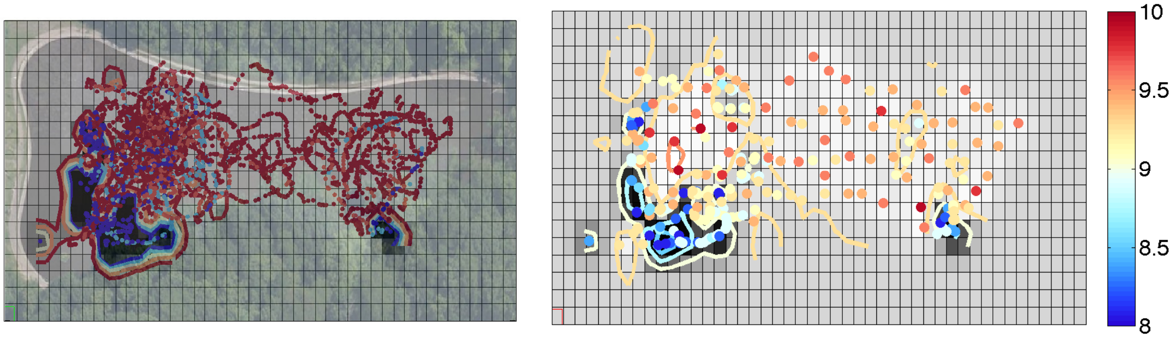
Precision plots are available in the appendix of Y. Ma, Sutherland, et al. (2015) for completeness. Precision is determined primarily by the classifier and θ, and thus does not vary much across methods.

7.4.1 Environmental monitoring (linear classifier)

In order to analyze the performance of APPS with the mean threshold classifier, we ran it on a real environmental monitoring dataset and compared to baseline algorithms. Valada et al. (2012)

[2] https://github.com/AutonlabCMU/ActivePatternSearch/

used small (60 cm) autonomous fan-powered boats to collect dissolved oxygen (do) readings in a pond, with the goal of finding regions that are low in dissolved oxygen, an indicator of poor water quality. The data used in our experiment comes from a pond approximately 150 meters wide and 50 meters long. The mobile robots have a cell-phone module that records the time and location of every measurement. Because of physical limitations, the measurement reading does not stabilize for about one minute. Therefore, in data collection, the boat was moved back and forth in a single location, in the hope that the noise would cancel by averaging these measurements.



(a) Data and true matching regions (black).

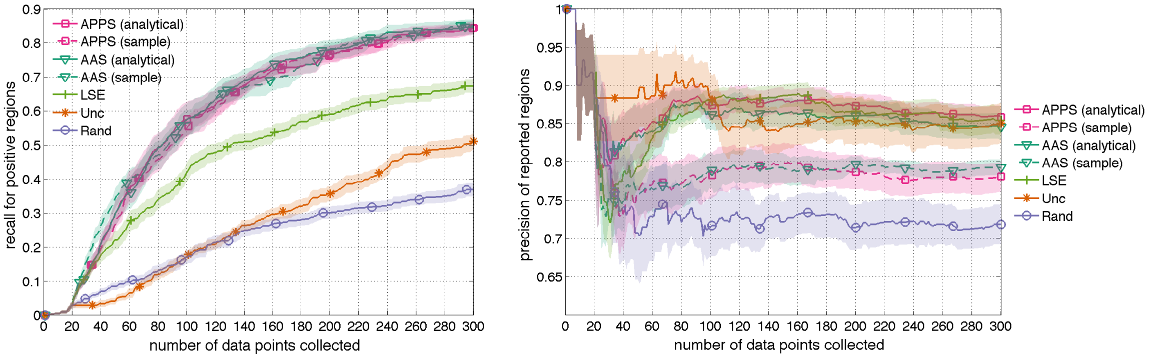
(b) apps collected data and posterior region probability.

Figure 7.1: Illustration of dataset and apps selections for one run. A point marks the location of a measurement whose value is also reflected in its color. Every grid box is a region whose possibility of matching is reflected in grayscale.

In order to verify our methods, we borrowed data from Valada et al. (2012), comprising 16 960 location/do value pairs, and fit a gp model by maximizing the likelihood of the prior parameters on 500 random samples seven times, taking the median of the learned hyperparameter values. We used a squared-exponential kernel with a learned length scale. We defined regions by covering the map with many windows of size comparable to the gp length scale, and used parameters $b = -9$, $c = -100$. Data points and classifier probability outputs for the ground truth are shown in Figure 7.1a, which also shows the learned length scale (roughly 3 meters).

We then repeated the following experiment: we randomly sampled 6 000 points at a time from data points not used for gp parameter training, and randomly selected 10 of these 6 000 points to form an initial training set $\mathcal{D}$. We then used several competing methods to sequentially make further queries until 300 total observations were obtained. The considered algorithms were: apps with analytical solutions, apps with one draw of z_* at each candidate location, aas (Y. Ma, Garnett, et al. 2014) with analytical solutions, aas with sampling, the level set estimation (lse) algorithm of Gotovos et al. (2013) with parameters $\beta^t = 6.25$ and $\varepsilon = 0.1$, uncertainty sampling (unc), and random selection (rand). Each algorithm chose queries based on its own criterion; the quality of queried points was evaluated by the mean threshold classifier with the above parameters and was then compared with true region labels that were computed by the mean threshold classifier using all 6 000 data points. A 70% marginal probability was chosen to be required for a region to be classified as matching ($\theta = 0.7$).

Figure 7.2a reports the mean and standard error of the recall of matching regions over 15 repetitions of this experiment. apps and aas with both analytical solutions and sampling performed equally well here. The similarity between APPS and AAS is also expected because in linear problems, the choice of c (the only difference between the algorithms here) is relatively minor. Notice that AAS is not able to handle any other classifier-based setting; this is the core contribution of APPS. To understand why analytical solutions were similar to sampling, notice that the data collection locations have to be constrained to those actually recorded, which makes it easier to obtain a near-optimal decision.



(a) Recall curves.

(b) Precision curves.

Figure 7.2: Results for the pond monitoring experiment. Color bands show standard errors after 15 runs.

The second group in performance ranking is the LSE method. We attempted to boost its performance by selecting its parameters to directly optimize the area under its recall curve, which was, in a sense, cheating. On further analysis of its query decisions, we saw LSE making, for the most part, qualitatively similar selection decisions to APPS. LSE will stop collecting data in a region if there is enough confidence, but does not specifically try to push regions over the threshold, and so its performance on this objective is inferior.

Last in the comparison are RAND and UNC. It is interesting to observe that RAND was initially better than, but later crossed by UNC. In the beginning, since UNC is purely explorative, its reward uniformly remained low across multiple runs, whereas in some runs RAND queries can be lucky enough to concentrate around matching regions. At a later phase, RAND faces the coupon collector's problem and may select redundant boring observations, whereas UNC keeps making progress at a constant rate.

Figure 7.2b shows results for precision. Sampling-based methods for APPS and AAS had lower precision than analytical ones, because the noise of sampling makes it more likely for an "accidental" flag of a region which then persists.

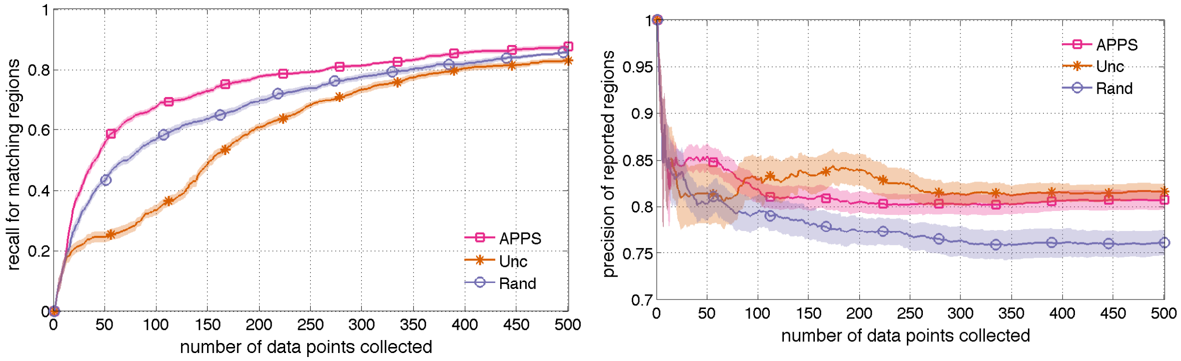
7.4.2 Predicting election results (linear classifier)

Consider the problem of a state-level political party official who wishes to determine which races will be won, lost, or might go either way. As surveying likely voters is relatively expensive, we would like to do so with as few surveys as possible.

In a simple model of this problem, the problem of finding races which will be won is a natural fit to a classifier of the form $h_g(f) = \Phi\big(w^\mathsf{T} f(\Xi_g) + b_g\big)$. Our function f maps from the voting precincts in the state to the vote share of a given party in that district, with a covariance kernel defined by demographic similarity and geographic proximity. To account for multiple races taking place in each district (e.g., state and national legislators), we duplicate each precinct with a flag for the type of election. If Ξ_g is the set of all precincts participating in a particular race and w_g is some constant c times the voting population of each precinct, then $w^\mathsf{T} f(\Xi_g)$ gives c times the total vote portion for the given party in that election. In a simple model which ignores turnout effects, the probability of winning a race is essentially 1 if the underlying proportion is greater than 0.5 and 0 otherwise; this can be accomplished by setting c to some fairly large constant, say 100, and $b = -\frac{1}{2}c$. (An equally simple model that nonetheless more thoroughly accounts for unmodeled effects would just use a smaller value of c.)

We ran experiments based on this model on 2010 Pennsylvania election returns (Ansolabehere and Rodden 2011). For each voting precinct in the dataset, we used the 2010 Decennial Census (United States Census Bureau 2010) to obtain a total population count and percentages of the population for gender, race, age, and housing type categories; we also added an (x, y) location based on a Lambert conformal conic projection of a point in the precinct, and used these features in a squared-exponential kernel. The data for each precinct was then replicated three times and associated with Democratic vote shares for its U.S. House of Representatives, Pennsylvania House of Representatives, and Pennsylvania State Senate races; the demographic/geographic kernel was multiplied by a positive-definite covariance matrix amongst the races. We learned the hyperparameters for this kernel by maximizing the likelihood of the model on full 2008 election data.

Given the kernel, we set up experiments to predict 2010 races based on surveying an individual voting precinct at a time. For simplicity, we assume that a given voting precinct can be thoroughly surveyed (and ignore turnout effects, voters changing their minds over time, and so on); thus observations were made with the true vote share. We seeded the experiment with a random 10 (out of 16 226) districts observed; APPS selected from a random subset of 100 proposals at each step. We again used $\theta = 0.7$.



(a) Recall curves.

(b) Precision curves.

Figure 7.3: Results for election prediction. Color bands show standard errors over 15 runs.

Figure 7.3a shows the mean and standard errors of recalls over 15 runs for APPS, UNC, and RAND. LSE and AAS are not applicable to this problem, as they have no notion of weighting points (by population). APPS outperforms both random and uncertainty sampling here, though in this case the margin over random sampling is much narrower. This is probably because the portion of regions which are positive in this problem is much higher, so more points are informative. Uncertainty sampling is in fact worse than random here, which is not too surprising because the purely explorative nature of UNC is even worse on the high dimensional input space of this problem.

7.4.3 Finding vortices (black-box classifier)

The problem we consider here requires more complex pattern classifiers. We study the task of identifying vortices in a vector field based on limited observations of flow vectors. Linear classifiers are insufficient for this problem,[3] so we will demonstrate the flexibility of our approach with a black-box classifier.

To illustrate this setting, we consider the results of a large-scale simulation of a turbulent fluid in three dimensions over time in the Johns Hopkins Turbulence Databases[4] (Perlman et al. 2007). Following Sutherland, Xiong, et al. (2012), we aim to recognize vortices in two-dimensional slices of the data at a single timestep, based on the same small training set of 11 vortices and 20 non-vortices, partially shown in Figure 7.4.

Recall that h_g assigns probability estimates to the entire function class $\mathcal{F}$ confined to region g. Unlike the previous examples, it is insufficient to consider only a weighted integral of f. We can consider the average flow across sectors (angular slices from the center) of our region as building blocks in detecting vortices. We count how many sectors have clockwise/counter-clockwise flows to give a classification result, in three steps:

1. First, we divide a region into K sectors. In each sector, we take the integral of the inner product between the actual flow vectors and a template. The template is an "ideal" vortex, but with larger weights in the center than the periphery. This produces a K-dimensional summary statistic $L_g(f)$ for each region.
2. Next, we improve robustness against different flow speeds in the data by scaling $L_g(f)$ to have maximum entry 1, and flip its sign if its mean is negative. Call the result $\tilde{L}_g(f)$.
3. Finally, we feed the normalized $\tilde{L}_g(f)$ vector through a 2-layer neural network of the form

   $$h_g(f) = \sigma\left(w_{\text{out}} \sum_{i=1}^{K} \sigma\left(w_{\text{in}} \tilde{L}_g(f)_i + b_{\text{in}}\right) + b_{\text{out}}\right),$$

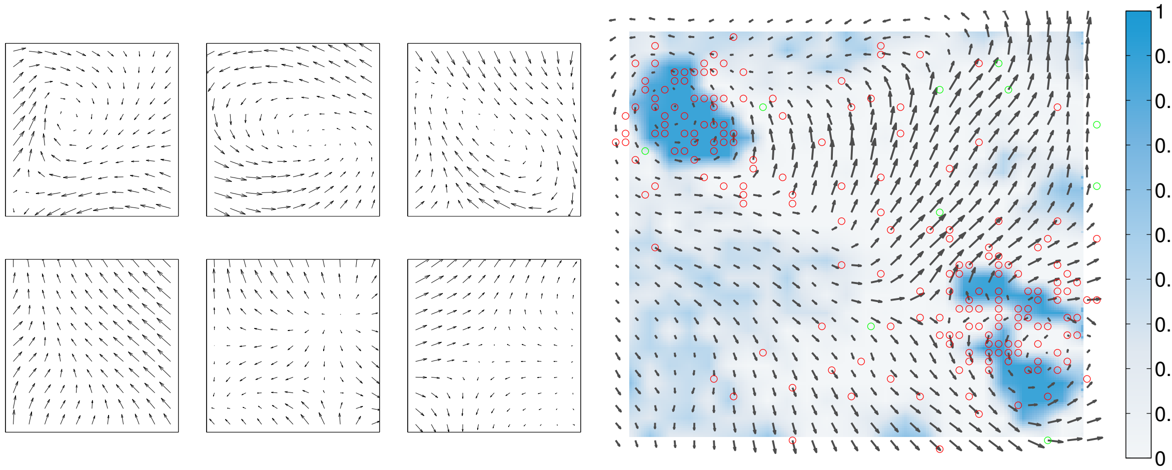
   where σ is the logistic sigmoid function.

Because L_g, which is effectively taking the L_2 inner product with K fixed template functions, is a linear operator, $L_g(f) \mid \mathcal{D}$ obeys a K-dimensional multivariate normal distribution. We sample many possible $L_g(f)$ from that distribution, which we then normalize and pass through

[3] The set of vortices is not convex: consider the midpoint between a clockwise vortex and its identical counter-clockwise case.

[4] http://turbulence.pha.jhu.edu

the neural network as described above. This gives samples of probabilities h_g, whose mean is a Monte Carlo estimate of (7.1).



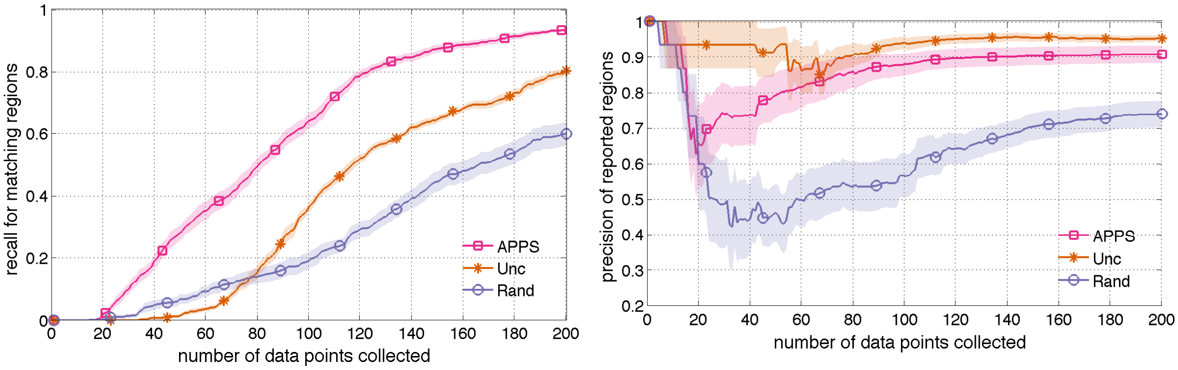
(a)

(b)

Figure 7.4: (a): Positive (top) and negative (bottom) training examples for the vortex classifier. (b): The velocity field used; each arrow is the average of a 2×2 square of actual data points. Background color shows the probability obtained by each region classifier on the 200 circled points; red circles mark points selected by one run of APPS initialized at the green circles.

We used $K = 4$ sectors, and the weights in the template were fixed such that the length scale matches the distance from the center to an edge. The network was optimized for classification accuracy on the training set. We then identified a 50×50-pixel slice of the data that contains two vortices, some other "interesting" regions, and some "boring" regions, mostly overlapping with Figure 11 of Sutherland, Xiong, et al. (2012); the region, along with the output of the classifier when given all of the input points, is shown in Figure 7.4a. We then ran APPS, initialized with 10 uniformly random points, for 200 steps. We defined the regions to be squares of size 11×11 and spaced them every 2 points along the grid, for 400 total regions. We again thresholded at $\theta = 0.7$. We evaluate (7.1) via a Monte Carlo approximation, as in the general form of the algorithm in Section 7.3: First, we pick 80 random candidate locations x_*. For each x_*, we took 4 samples of z_*. For each z_*, we obtained the posterior of f over the evaluation window, and evaluated h_g on 15 different samples from that posterior.

Figure 7.5a shows recall curves of APPS, uncertainty sampling (UNC), and random selection (RAND), where for the purpose of these curves we call the true label the output of the classifier when all data is known, and the proposed label is true if $T_g > \theta$ at that point of the search (evaluated using more Monte Carlo samples than in the search process, to gain assurance in our evaluation but without increasing the time required for the search). We can see that active pattern search substantially outperforms uncertainty sampling and random selection. It is interesting to observe that RAND was initially better than, but later crossed by UNC. In the beginning, since UNC is purely explorative, its reward uniformly remained low across multiple runs, whereas in some runs RAND queries can be lucky enough to concentrate around matching regions. At a later



(a) Recall curves.

(b) Precision curves.

Figure 7.5: Results for the vortex experiment. Color bands show standard errors over 15 runs.

phase, RAND faces the coupon collector's problem and may select redundant boring observations, whereas UNC keeps making progress at a constant rate.

Chapter 8

Conclusions and future directions

If there is but a single take-away message from this thesis, it is perhaps that: in any machine learning problem, it is vital to consider how to model your data. Sets and distributions are a flexible choice that cover many use cases, and can be applied to many different problem areas, as seen in Chapters 1 and 5.

Random feature embeddings are also an important tool for scalable learning, but when using random Fourier features make sure to use the right choice (Chapter 3). They have important advantages over the Nyström approach in ease of distributing across multiple machines and of integration into deep learning settings; their relative performance depends on the problem setting, but Nyström embeddings with approximate leverage scores seem promising (T. Yang et al. 2012; El Alaoui and Mahoney 2015; Rudi et al. 2016). For distribution learning, the MMD embedding based on random Fourier features for the Gaussian RBF kernel (Section 4.1) is very simple and typically performs well, though in some cases the HDD embedding of Section 4.3 may be better.

For hard problems, learning more complex kernels is extremely important. We have proposed a promising new method for doing so in the setting of two-sample testing in Chapter 6, and its integration with deep learning to learn very powerful kernels is quite promising. For general learning on distributions, integration with deep networks as proposed in Section 8.2 is a promising way forward that still needs more study.

Active learning is also an important problem with many real-world applications. Chapter 7 gave an algorithm for the particular problem of active pointillistic pattern search, which is of a similar flavor to learning on distributions. Section 8.4 discusses some approaches to true active learning on distributions.

How to solve a new distribution learning problem Given a new problem that can reasonably be phrased as a distribution learning problem, the "default" choice should probably be with MMD based on the Gaussian RBF kernel, which is the simplest approach that has shown empirical success in a variety of areas and is supported by theory (Szabó et al. 2015); either the pairwise estimator or the random Fourier feature embedding is fine, though for either large numbers of distributions or for many samples from each distribution the embedding is quite helpful. Tuning the kernel bandwidth is important, and should probably be done by cross-validation on the final learning performance.

If performance there is not satisfactory, for moderate sample sizes pairwise estimators of other distributional distances (as in Chapter 2) may work better; for larger sample sizes with low-dimensional distributions, the embedding of Section 4.3 is sometimes preferable. With high-dimensional distributions and large sample sizes, perhaps the dimensions can be treated independently (as in Section 5.3.2), but otherwise these basic choices are exhausted.

If better performance is still required, the best choice is probably to explore integration of the MMD embedding with deep learning, as in Section 8.2.

The remainder of this chapter discusses future areas to explore.

8.1 Deep learning of kernels for two-sample testing

The t_U and τ_U statistics of Chapter 6 are quite naturally suited to performing two-sample testing problems with deep learning. Thorough evaluations of this approach to difficult two-sample testing problems are underway now.

One prominent potential application is to the very popular framework of generative adversarial networks (Goodfellow et al. 2014), in which a generator network attempts to create samples that look like training samples by tricking an adversary network, which attempts to distinguish generated samples from training. As simultaneously noted by Y. Li et al. (2015) and Dziugaite et al. (2015), the adversary network can be thought of as performing a two-sample test between a batch of generated samples and the training set, and so the adversary can be simply replaced by an MMD test. Dziugaite et al. (2015) attempted to do so with a fixed Gaussian RBF kernel, which performed poorly on generating images, because the kernel has a very poor understanding of images. Y. Li et al. (2015) worked around this by (essentially) using MMD with the Gaussian RBF kernel on the latent codes learned by a fixed autoencoder instead, and got much better results. We may be able to do better, however, by using an adversary based on an MMD test with a kernel learned (via the t_U or τ_U criteria of Chapter 6) for the particular comparison at hand. This method will be fully adaptive to whatever the generator network chooses to create, rather than relying on the fixed autoencoder-based kernel as in Y. Li et al. (2015).

8.2 Deep learning of kernels for distribution learning

Manually specifying featurizations and kernels can be an arduous task, especially for those inexperienced with the precise methods in use. In certain problems in computer vision, even years of extremely active development on different human-designed featurizations have not matched the performance of learned features. The further adoption of distribution learning would benefit greatly from integration with representation learning techniques.

In Chapter 6, we explored the automated learning of complex kernels for two-sample testing problems, primarily using pairwise kernel estimators. Though similar techniques could be applicable to regression and classification tasks — and indeed Yoshikawa et al. (2014, 2015) use techniques that could be viewed as being along these lines — when there are many distributions to compare, rather than just the two of a two-sample testing problem, that task is more difficult.

The embeddings discussed in Chapter 4, however, provide a natural means to use deep learning techniques in distribution learning, and vice versa: given inputs $\{x_1, \dots, x_n\}$, simply compute a deep representation $\{f(x_1), \dots, f(x_n)\}$ convolutionally, then pass through $z(\{f(x_1), \dots, f(x_n)\})$ before performing the learning task. The MMD embedding of Section 4.1 and the L_2 embedding of Section 4.2 are both easily differentiable (depending on the choice of kernel or basis functions), and simple to implement within deep learning frameworks. The HDD embeddings of Section 4.3 would be more complex, though possible, to use in this manner.

8.2.1 Integration with deep computer vision models

In Section 5.3.2, we considered using the features learned by a standard convolutional deep network as samples from an image-level distribution of local features, and classified images based on those sets of features. Here features are trained using fully-connected final layers as the learning model, but then used in a separate distributional kernel model. We can instead make a coherent model which combines feature extraction with a learning model based on a distributional kernel, by making a distributional embedding layer in the network.

In fact, the "network in network" architecture of M. Lin et al. (2014) popularized the idea of replacing the late-layer fully-connected layers of AlexNet-type models (Krizhevsky et al. 2012) with global average pooling, treating each convolutional filter as providing a score for a given class label and aggregating with the mean. Szegedy et al. (2014) later adopted this idea, though they added a layer after the average pooling in an attempt to ease cross-task adaptation. In the distributional framework, we can think of this now as a classifier based on a linear-kernel MMD embedding.

Linear-kernel MMD, however, compares distributions based only on their mean. By using e.g. random Fourier features for a Gaussian RBF kernel, we can derive a richer classifier structure. We conducted an initial exploration of this approach in J. B. Oliva, Sutherland, et al. (2015), taking networks initially trained on ImageNet (Russakovsky et al. 2014) and adapting them to other classification tasks: Flickr Style (Karayev et al. 2013), Wikipaintings (Karayev et al. 2013), and Places (Zhou et al. 2014). We used both AlexNet and GoogLeNet architectures, either replacing or augmenting the final classification layers with random Fourier feature Gaussian RBF embeddings, and found small but consistent improvements in classification accuracies compared to adapting the original model. For details, see J. B. Oliva, Sutherland, et al. (2015).

It seems plausible that the reason the improvements here were not as large as we might have hoped is that we were fine-tuning features initially found to work for the existing architecture, whereas the optimal features to use when making full distributional comparisons are probably somewhat different. Thus, training the distributional variants of the network from scratch and perhaps varying the earlier architecture of the network as well would be required to fully realize the potential of this work. We leave this time-consuming process to future work.

8.2.2 Other paramaterizations for kernel learning

In addition to learning the mapping $f(x)$ used before the kernel, one can also consider learning the kernel itself. When using a random Fourier feature-based approach, learning the bandwidth of the kernel is simple: sample $\omega_i \sim \mathcal{N}(0, I_d)$ and then scale the inputs by $\sigma \omega_i$, perhaps with σ

parameterized as $\sigma = \exp(s)$. However, one can also consider learning the values ω_i themselves, learning the kernel via its Fourier transform. This was evaluated in J. B. Oliva, Sutherland, et al. (2015).

This parameterization, however, might not be the best way to learn the kernel. Z. Yang et al. (2015) use the Fastfood approximation (Le et al. 2013) to random Fourier features and learn only certain parts of the spectral representation, rather than directly adjusting the frequencies. This may result in a nicer optimization surface.

8.3 Word and document embeddings as distributions

Until recently, much work in natural language processing treated words as unique symbols, e.g. with "one-hot" vectors, where the ith word from a vocabulary of size V is represented as a vector with ith component 1 and all other components 0. It has recently become widely accepted that applications can benefit from richer word embeddings which take into account the similarity between distinct words, and much work has been done on dense *word embeddings* so that distances or inner products between word embeddings represent word similarity in some way (e.g. Collobert and Weston 2008; Turian et al. 2010; Mikolov et al. 2013). These embeddings can be learned in various ways, but often involve optimizing the representation's performance in some supervised learning task.

Document representations First, it is worth noting that although this breaks the traditional "bag of words" text model (where documents can be represented simply by the sum of the words' one-hot encodings), we can represent documents by viewing them as sample sets of word vectors.

Kusner et al. (2015) recently adopted this model, using kNN classifiers based on the Earth Mover's Distance (EMD) between documents, and obtained excellent empirical results. EMD, however, is expensive to compute even for each pair of documents when the vocabulary is large, and additionally must be computed pairwise between documents; an approximate embedding in the style of Chapter 4 is not known.

Yoshikawa et al. (2014), in their empirical results, considered this model with MMD-based kernels (but computing pairwise kernel values rather than approximate embeddings). Their main contribution, however, is to optimize the word embedding vectors for final classification performance; by doing so with random initializations, they saw mild performance improvements over MMD kernels using substantially less training data for the embeddings but at much higher computational cost. Yoshikawa et al. (2015) extend the approach to Gaussian process regression models, but do not compare to separately-learned word embeddings.

Because of the limited empirical evaluation, particularly on larger datasets, it is currently unclear how these methods compare to one another or to other approaches for document representation. Additionally, perhaps fine-tuning existing word embeddings learned on a standard dataset simultaneously with learning the regression or classification model for a particular application, as is common in deep learning models for computer vision, would provide additional power.

Richer word representation Embedding words as a single vector does not allow for as rich a word representation as we might wish. Vilnis and McCallum (2015) embed words instead as Gaussian distributions, and use the KL divergence between word embeddings to measure asymmetric hypernym relationships: for example, their embedding for the word *Bach* is "included" in their embeddings for *famous* and *man*, and mostly included in *composer*. Gaussian distributions, of course, are still fairly limiting; for example, a multimodal embedding might be able to capture word sense ambiguity, whereas a Gaussian embedding would be forced to attempt to combine both senses in a single broad embedding.

We can thus consider richer, nonparametric classes of word embeddings: perhaps by representing a word as a (possibly weighted) set of latent vectors. Comparisons could then be performed either with an MMD-based kernel, when symmetry is desired, or with KL estimators (or similar) when not.

One approach would be to choose these vectors arbitrarily, optimizing them for the output of some learning problem: this would be implemantionally similar to the approach of Yoshikawa et al. (2014, 2015) for MMD distances, or somewhat like that of Vilnis and McCallum (2015) but with greater computational cost, and greater flexibility, for KL distances.

Another approach is inspired by the classic distributional hypothesis of Harris (1954), that the semantics of words are characterized by the contexts in which it appears. Many word embedding approaches can be viewed as matrix factorizations of a matrix M with rows corresponding to words, columns to some notion of context, and entries containing some measure of association between the two; the factorization $M = WC^\mathsf{T}$ then typically discards the matrix C and uses the rows of W as word vectors. This approach is sometimes taken explicitly; interestingly, the popular method of Mikolov et al. (2013) can be seen as approximating this form as well (Levy and Goldberg 2014). This view inspires a natural alternative: treat each word as the sample set of contexts in which it appears, representing each context via the learned context vectors. This is perhaps the most direct instantiation of the distributional hypothesis: compare words by comparing the distribution of contexts in which they appear.

8.4 Active learning on distributions

Suppose we have a collection of distributions, but initially we have very few samples from each distribution. We can choose to take additional iid observations, but doing so is relatively expensive; perhaps it requires real-world expenditure of time or resources to collect samples, or perhaps these distributions are available only through computationally intensive numerical simulations. We may wish to learn a classification or regression function mapping from these distributions to some label (similar to traditional active learning settings), to locate distributions which follow some prespecified pattern (similar to the setting of Chapter 7 with independent regions), or to find the distribution which is "best" in some sense (as in pure-exploration bandit problems, Bubeck et al. 2010). In any of these cases, we need to choose some selection criterion that will appropriately consider the utility of selecting points from distributions, a problem that is related to but certainly distinct from typical fully-observed active learning models.

In the dark matter prediction experiments of Section 5.1, we assumed that each observed galaxy has a well-known line-of-sight velocity estimated via redshift. In practice, good velocity estimates are available only through relatively-expensive spectroscopic imaging; cheaper few-color imaging techniques give extremely uncertain velocity estimates. We could simply ignore the imaging estimates and apply the previous model, selecting a random galaxy from each halo to perform spectroscopy upon. It would probably be more effective, however, to consider active learning methods that begin with visual imaging, and then identify which objects will be useful for spectroscopy in order to best identify the masses of their dark matter halos. One modeling option would be to take a probability distribution over the sample set, and then identify the resulting distribution of the mean map embedding and therefore its predicted label under a learned predictor; we would then identify objects to observe that most reduce uncertainty in the predicted label. This could be conducted either for a single halo, where the objective is to best learn its mass, or across multiple halos, where the objective is either to find the most massive halos (active search) or to reduce some form of overall uncertainty in all of the halo mass predictions (active learning).

Appendix A

The `skl-groups` package

Efficient implementations of several of the methods for learning on distributions discussed in this thesis are available in the Python package `skl-groups`[1]. This package integrates with the standard Python numerical ecosystem and presents an API compatible with that of `scikit-learn` (Grisel et al. 2016).

The package is designed around the `Transformer` interface of `scikit-learn`, and as much as possible to work with its pipelines. But the `scikit-learn` API, which works almost exclusively with Euclidean feature vectors, it is assumed in most places that features are represented as an array of shape (n, d), where each row represents a feature vector. In `skl-groups`, each object is a set of vectors. This is represented as a Python `list` (or numpy `object` array) of numeric arrays, where each array is of shape (n_i, d): n_i can vary from element to element. Internally, most methods convert these leasts into `Features` objects, which provide convenient helpers to access the data in a consistent way, and can optionally store any metadata associated with each element.

The class supports data storage either as a collection of separate numeric arrays, or as a single stacked array of shape $(\sum_i n_i, d)$, with views into the array for each feature set. This form is convenient for more efficient memory access or for operations which operate pointwise (like standardization).

The `skl-groups` API can be divided into several sections:

Features The `Features` class discussed above.

Preprocessing A collection of utilities to normalize, scale, standardize, or run principal components analysis on each set in a collection of features. These are wrappers around a class `BagPreprocesser`, which helps apply transformers to each set, and the relevant `scikit-learn` transformers.

Summaries Methods that convert sets into single feature vectors:

- `BagMean`: Represents each set by its mean. Especially useful in conjunction with `scikit-learn`'s `RBFSampler` to perform the MMD embedding of Section 4.1.
- `BagOfWords`: Quantizes each set into the bag of words representation.

[1] https://github.com/dougalsutherland/skl-groups

- `L2DensityTransformer`: The L_2 embedding of Section 4.2.

Set kernels Currently contains only `MeanMapKernel`, which computes the pairwise MMD estimator.

Divergences `KNNDivergenceEstimator`, which can estimate $D_{\alpha,\beta}$ divergences based on Póczos and Schneider (2011) and Póczos, Xiong, Sutherland, et al. (2012), the KL divergence based on Q. Wang et al. (2009), and an estimator for the Jensen-Shannon divergence based on Hino and Murata (2013).

Kernel utilities Utilities to turn a divergence into an RBF kernel, as well as the PSD corrections of Section 2.4.1.

Miscellaneous utilities Utilities to show a progress bar for long-running operations like the k-NN divergence estimator.

Appendix B

Proofs for Chapter 3

B.1 Proof of Proposition 3.4

Part (i) is particularly simple:

$$\begin{aligned}
\mathbb{E}\|\tilde{f}\|_\mu^2 &= \mathbb{E}\int_{\mathcal{X}^2} \tilde{f}(x,y)^2 \,\mathrm{d}\mu(x,y) \\
&= \int_{\mathcal{X}^2} \mathbb{E}\ \tilde{f}(x,y)^2 \,\mathrm{d}\mu(x,y) \qquad \text{(B.1)} \\
&= \int_{\mathcal{X}^2} \frac{1}{D}\left[1 + k(2x,2y) - 2k(x,y)^2\right] \mathrm{d}\mu(x,y)
\end{aligned}$$

where (B.1) is justified by Tonelli's theorem.

For part (ii), view $\|\tilde{f}\|_\mu$ as a function of $\omega_1, \ldots, \omega_{D/2}$; then changing ω_i to a different $\hat{\omega}_i$ changes the value of $\|\tilde{f}\|_\mu$ by at most $8\frac{2D+1}{D^2}\mu(\mathcal{X}^2)$ (as will be shown shortly). The first inequality is thus a direct application of McDiarmid (1989); the second simply notes that $\frac{D^2}{32(2D+1)^2} \geq \frac{1}{288}$.

To show the claimed bounded deviation property, assume without loss of generality that we replace ω_1 by $\hat{\omega}_1$:

$$\begin{aligned}
&\left|\|\tilde{f}\|_\mu^2(\omega_1, \omega_2, \ldots, \omega_{D/2}) - \|\tilde{f}\|_\mu^2(\hat{\omega}_1, \omega_2, \ldots, \omega_{D/2})\right| \\
&\quad= \Bigg|\int_{\mathcal{X}^2}\left(\frac{2}{D}\cos(\omega_1^\mathsf{T}(x-y)) + \frac{2}{D}\sum_{i=2}^{D/2}\cos(\omega_i^\mathsf{T}(x-y)) - k(x,y)\right)^2 \mathrm{d}\mu(x,y) \\
&\qquad - \int_{\mathcal{X}^2}\left(\frac{2}{D}\cos(\hat{\omega}_1^\mathsf{T}(x-y)) + \frac{2}{D}\sum_{i=2}^{D/2}\cos(\omega_i^\mathsf{T}(x-y)) - k(x,y)\right)^2 \mathrm{d}\mu(x,y)\Bigg| \\
&\quad= \Bigg|\frac{4}{D^2}\int_{\mathcal{X}^2}\cos^2(\omega_1^\mathsf{T}(x-y))\mathrm{d}\mu(x,y) + \int_{\mathcal{X}^2}\left(\frac{2}{D}\sum_{i=2}^{D/2}\cos(\omega_i^\mathsf{T}(x-y)) - k(x,y)\right)^2 \mathrm{d}\mu(x,y) \\
&\qquad +2\int_{\mathcal{X}^2}\frac{2}{D}\cos(\omega_1^\mathsf{T}(x-y))\left(\frac{2}{D}\sum_{i=2}^{D/2}\cos(\omega_i^\mathsf{T}(x-y)) - k(x,y)\right)\mathrm{d}\mu(x,y)
\end{aligned}$$

$$
\begin{aligned}
&-\frac{4}{D^2}\int_{\mathcal{X}^2}\cos^2(\hat{\omega}_1^\mathsf{T}(x-y))\mathrm{d}\mu(x,y)-\int_{\mathcal{X}^2}\left(\frac{2}{D}\sum_{i=2}^{D/2}\cos(\omega_i^\mathsf{T}(x-y))-k(x,y)\right)^2\mathrm{d}\mu(x,y)\\
&-2\int_{\mathcal{X}^2}\frac{2}{D}\cos(\hat{\omega}_1^\mathsf{T}(x-y))\left(\frac{2}{D}\sum_{i=2}^{D/2}\cos(\omega_i^\mathsf{T}(x-y))-k(x,y)\right)\mathrm{d}\mu(x,y)\Bigg|\\
=&\Bigg|\frac{4}{D^2}\int_{\mathcal{X}^2}\left(\cos^2(\omega_1^\mathsf{T}(x-y))-\cos^2(\hat{\omega}_1^\mathsf{T}(x-y))\right)\mathrm{d}\mu(x,y)\\
&+\frac{4}{D}\int_{\mathcal{X}^2}\left(\cos(\omega_1^\mathsf{T}(x-y))-\cos(\hat{\omega}_1^\mathsf{T}(x-y))\right)\left(\frac{2}{D}\sum_{i=2}^{D/2}\cos(\omega_i^\mathsf{T}(x-y))-k(x,y)\right)\mathrm{d}\mu(x,y)\Bigg|\\
\leq&\frac{4}{D^2}\int_{\mathcal{X}^2}2\mathrm{d}\mu(x,y)+\frac{4}{D}\int_{\mathcal{X}^2}4\mathrm{d}\mu(x,y)\\
=&\left(\frac{8}{D^2}+\frac{16}{D}\right)\mu(\mathcal{X}^2)=\frac{16D+8}{D^2}\mu(\mathcal{X}^2).
\end{aligned}
$$

B.2 Proof of Proposition 3.5

Part (i) is exactly analagous to that for $\tilde{f}$. Part (ii) is also quite similar:

$$
\begin{aligned}
&\left|\|\breve{f}\|_\mu^2(\omega_1,\omega_2,\ldots,\omega_{D/2})-\|\breve{f}\|_\mu^2(\hat{\omega}_1,\omega_2,\ldots,\omega_{D/2})\right|\\
&\quad=\Bigg|\int_{\mathcal{X}^2}\Bigg(\frac{2}{D}\left(\cos(\omega_1^\mathsf{T}(x-y))+\cos(\omega_1^\mathsf{T}(x+y)+2b_1)\right)\\
&\qquad\qquad+\frac{2}{D}\sum_{i=2}^{D/2}\left[\cos(\omega_i^\mathsf{T}(x-y))+\cos(\omega_i^\mathsf{T}(x+y)+2b_i)\right]-k(x,y)\Bigg)^2\mathrm{d}\mu(x,y)\\
&\qquad-\int_{\mathcal{X}^2}\Bigg(\frac{2}{D}\left(\cos(\hat{\omega}_1^\mathsf{T}(x-y))+\cos(\hat{\omega}_1^\mathsf{T}(x+y)+2b_1)\right)\\
&\qquad\qquad+\frac{2}{D}\sum_{i=2}^{D/2}\left[\cos(\omega_i^\mathsf{T}(x-y))+\cos(\omega_i^\mathsf{T}(x-y)+2b_i)\right]-k(x,y)\Bigg)^2\mathrm{d}\mu(x,y)\Bigg|\\
&\quad=\Bigg|\frac{4}{D^2}\int_{\mathcal{X}^2}\left(\left(\cos(\omega_1^\mathsf{T}(x-y))+\cos(\omega_1^\mathsf{T}(x+y)+2b_i)\right)^2-\left(\cos(\hat{\omega}_1^\mathsf{T}(x-y))+\cos(\hat{\omega}_1^\mathsf{T}(x+y)+2b_i)\right)^2\right)\mathrm{d}\mu(x,\\
&\qquad+\frac{4}{D}\int_{\mathcal{X}^2}\left(\cos(\omega_1^\mathsf{T}(x-y))+\cos(\omega_1^\mathsf{T}(x+y)+2b_i)-\cos(\hat{\omega}_1^\mathsf{T}(x-y))-\cos(\hat{\omega}_1^\mathsf{T}(x+y)+2b_i)\right)\\
&\qquad\qquad\left(\frac{2}{D}\sum_{i=2}^{D/2}\left[\cos(\omega_i^\mathsf{T}(x-y))+\cos(\omega_i^\mathsf{T}(x+y)+2b_i)\right]-k(x,y)\right)\mathrm{d}\mu(x,y)\Bigg|\\
&\quad\leq\frac{4}{D^2}\int_{\mathcal{X}^2}8\,\mathrm{d}\mu(x,y)+\frac{4}{D}\int_{\mathcal{X}^2}4\times3\,\mathrm{d}\mu(x,y)
\end{aligned}
$$

$$= \frac{32}{D^2}\mu(\mathcal{X}^2) + \frac{48}{D}\mu(\mathcal{X}^2)$$
$$= 16\frac{3D+2}{D^2}\mu(\mathcal{X}^2).$$

B.3 Proof of Proposition 3.6

The proof strategy closely follows that of Rahimi and Recht (2007); we fill in some (important) details, tightening some parts of the proof as we go.

Let $\mathcal{X}_\Delta = \{x - y \mid x, y \in \mathcal{X}\}$. It's compact, with diameter at most 2ℓ, so we can find an ε-net covering $\mathcal{X}_\Delta$ with at most $T = (4\ell/r)^d$ balls of radius r (Cucker and Smale 2001, Proposition 5). Let $\{\Delta_i\}_{i=1}^T$ denote their centers, and $L_{\tilde{f}}$ be the Lipschitz constant of $\tilde{f}$. If $|\tilde{f}(\Delta_i)| < \varepsilon/2$ for all i and $L_{\tilde{f}} < \varepsilon/(2r)$, then $|\tilde{f}(\Delta)| < \varepsilon$ for all $\Delta \in \mathcal{X}_\Delta$.

Let $\tilde{z}_i(x) := \left[\sin(\omega_i^\mathsf{T} x) \quad \cos(\omega_i^\mathsf{T} x)\right]^\mathsf{T}$, so that $\tilde{z}(x)^\mathsf{T}\tilde{z}(y) = \frac{1}{D/2}\sum_{i=1}^{D/2} \tilde{z}_i(x)^\mathsf{T}\tilde{z}_i(y)$.

B.3.1 Regularity Condition

We will first need to establish that $\mathbb{E}\,\nabla\tilde{s}(\Delta) = \nabla\,\mathbb{E}\,\tilde{s}(\Delta) = \nabla k(\Delta)$. This can be proved via the following form of the Leibniz rule, quoted verbatim from Cheng (2013):

Theorem (Cheng 2013, Theorem 2)**.** *Let X be an open subset of $\mathbb{R}$, and Ω be a measure space. Suppose $f : X \times \Omega \to \mathbb{R}$ satisfies the following conditions:*

1. *$f(x, \omega)$ is a Lebesgue-integrable function of ω for each $x \in X$.*
2. *For almost all $\omega \in \Omega$, the derivative $\frac{\partial f(x,\omega)}{\partial x}$ exists for all $x \in X$.*
3. *There is an integrable function $\Theta : \Omega \to \mathbb{R}$ such that $\left|\frac{\partial f(x,\omega)}{\partial x}\right| \le \Theta(\omega)$ for all $x \in X$.*

Then for all $x \in X$,

$$\frac{\mathrm{d}}{\mathrm{d}x}\int_\Omega f(x,\omega)\,\mathrm{d}\omega = \int_\Omega \frac{\partial}{\partial x} f(x,\omega)\,\mathrm{d}\omega.$$

Define the function $\tilde{g}_{x,y}^i : \mathbb{R} \times \Omega \to \mathbb{R}$ by $\tilde{g}_{x,y}^i(t,\omega) = \tilde{s}_\omega(x + te_i, y)$, where e_i is the ith standard basis vector, and $\vec{\omega}$ is the tuple of all the ω_i used in $\tilde{z}$. $\tilde{g}_{x,y}^i(t,\cdot)$ is Lebesgue integrable in ω, since

$$\int \tilde{g}_{x,y}^i(t,\omega)\,\mathrm{d}\omega = \mathbb{E}\,\tilde{s}(x + te_i, y) = k(x + te_i, y) < \infty.$$

For any $\omega \in \Omega$, $\frac{\partial}{\partial t}\tilde{g}_{x,y}^i(t,\omega)$ exists, and satisfies:

$$\begin{aligned}\mathbb{E}_\omega\left|\frac{\partial}{\partial t} g_{x,y}^i(t,\omega)\right| &= \mathbb{E}_\omega\left|\frac{2}{D}\sum_{j=1}^{D/2}\sin(\omega_j^\mathsf{T} y)\frac{\partial}{\partial t}\sin(\omega_j^\mathsf{T} x + t\omega_{ji}) + \cos(\omega_j^\mathsf{T} y)\frac{\partial}{\partial t}\cos(\omega_j^\mathsf{T} x + t\omega_{ji})\right| \\ &= \mathbb{E}_\omega\left|\frac{2}{D}\sum_{j=1}^{D/2}\omega_{ji}\sin(\omega_j^\mathsf{T} y)\cos(\omega_j^\mathsf{T} x + t\omega_{ji}) - \omega_{ji}\cos(\omega_j^\mathsf{T} y)\sin(\omega_j^\mathsf{T} x + t\omega_{ji})\right|\end{aligned}$$

$$\begin{aligned}
&\leq \mathbb{E}_\omega \left[\frac{2}{D} \sum_{j=1}^{D/2} \left| \omega_{ji} \sin(\omega_j^\mathsf{T} y) \cos(\omega_j^\mathsf{T} x + t\omega_{ji}) \right| + \left| \omega_{ji} \cos(\omega_j^\mathsf{T} y) \sin(\omega_j^\mathsf{T} x + t\omega_{ji}) \right| \right] \\
&\leq \mathbb{E}_\omega \left[\frac{2}{D} \sum_{j=1}^{D/2} 2 \left| \omega_{ji} \right| \right] \\
&\leq 2\, \mathbb{E}_\omega \left| \omega_1 \right|,
\end{aligned}$$

which is finite since the first moment of ω_1 is assumed to exist.

Thus we have $\frac{\partial}{\partial x_i} \mathbb{E}\, \tilde{s}(x, y) = \mathbb{E}\, \frac{\partial}{\partial x_i} \tilde{s}(x, y)$. The same holds for y by symmetry. Combining the results for each component, we get as desired that $\mathbb{E}\, \nabla_\Delta s(x, y) = \nabla_\Delta\, \mathbb{E}\, s(x, y)$.

B.3.2 Lipschitz Constant

Since $\tilde{f}$ is differentiable, $L_{\tilde{f}} = \left\| \nabla \tilde{f}(\Delta^*) \right\|$, where $\Delta^* = \operatorname{argmax}_{\Delta \in \mathcal{X}_\Delta} \left\| \nabla \tilde{f}(\Delta) \right\|$.

Via Jensen's inequality, $\mathbb{E}\, \|\nabla \tilde{s}(\Delta)\| \geq \|\mathbb{E}\, \nabla \tilde{s}(\Delta)\|$. Now, letting $\Delta^* = x^* - y^*$:

$$\begin{aligned}
\mathbb{E}[L_{\tilde{f}}^2] &= \mathbb{E}\left[\|\nabla \tilde{s}(\Delta^*) - \nabla k(\Delta^*)\|^2 \right] \\
&= \mathbb{E}_{\Delta^*} \left[\mathbb{E}\left[\|\nabla \tilde{s}(\Delta^*)\|^2 \right] - 2\, \|\nabla k(\Delta^*)\|\, \mathbb{E}\left[\, \|\nabla \tilde{s}(\Delta^*)\|\, \right] + \|\nabla k(\Delta^*)\|^2 \right] \\
&\leq \mathbb{E}_{\Delta^*} \left[\mathbb{E}\left[\|\nabla \tilde{s}(\Delta^*)\|^2 \right] - 2\, \|\nabla k(\Delta^*)\|^2 + \|\nabla k(\Delta^*)\|^2 \right] \\
&= \mathbb{E}\left[\|\nabla \tilde{s}(\Delta^*)\|^2 \right] - \mathbb{E}_{\Delta^*}\left[\|\nabla k(\Delta^*)\|^2 \right] \\
&\leq \mathbb{E}\, \|\nabla \tilde{s}(\Delta^*)\|^2 \\
&= \mathbb{E} \left\| \nabla \tilde{z}(x^*)^\mathsf{T} \tilde{z}(y^*) \right\|^2 \\
&= \mathbb{E} \left\| \nabla \frac{1}{D/2} \sum_{i=1}^{D/2} \tilde{z}_i(x^*)^\mathsf{T} \tilde{z}_i(y^*) \right\|^2 \\
&= \mathbb{E} \left\| \nabla \tilde{z}_i(x^*)^\mathsf{T} \tilde{z}_i(y^*) \right\|^2 \qquad \text{(B.2)} \\
&= \mathbb{E} \left\| \nabla \cos(\omega^\mathsf{T} \Delta^*) \right\|^2 \\
&= \mathbb{E} \left\| - \sin(\omega^\mathsf{T} \Delta^*)\, \omega \right\|^2 \\
&= \mathbb{E} \left[\sin^2(\omega^\mathsf{T} \Delta^*)\, \|\omega\|^2 \right] \\
&\leq \mathbb{E}\left[\|\omega\|^2 \right] = \sigma_p^2.
\end{aligned}$$

We can thus use Markov's inequality:

$$\Pr\left(L_{\tilde{f}} \geq \frac{\varepsilon}{2r} \right) = \Pr\left(L_{\tilde{f}}^2 \geq \left(\frac{\varepsilon}{2r} \right)^2 \right) \leq \sigma_p^2 \left(\frac{2r}{\varepsilon} \right)^2.$$

B.3.3 Anchor Points

For any fixed $\Delta = x - y$, $\tilde{f}(\Delta)$ is a mean of $D/2$ terms with mean 0 and bounded by ± 1. Applying Hoeffding's inequality and a union bound:

$$\Pr\left(\bigcup_{i=1}^{T} |\tilde{f}(\Delta_i)| \geq \tfrac{1}{2}\varepsilon\right) \leq T \Pr\left(|\tilde{f}(\Delta)| \geq \tfrac{1}{2}\varepsilon\right) \leq 2T \exp\left(-\frac{2\frac{D}{2}\left(\frac{\varepsilon}{2}\right)^2}{(1-(-1))^2}\right) = 2T \exp\left(-\frac{D\varepsilon^2}{16}\right).$$

Alternatively, since we know from (3.4) that the variance of each term is $\operatorname{Var}[\cos(\omega^\mathsf{T}\Delta)] = \frac{1}{2} + \frac{1}{2}\underline{k}(2\Delta) - \underline{k}(\Delta)^2$, we could use Bernstein's inequality:

$$\begin{aligned} T \Pr\left(\left|\tilde{f}(\Delta)\right| > \tfrac{1}{2}\varepsilon\right) &\leq 2T \exp\left(-\frac{\frac{D}{2}\frac{\varepsilon^2}{4}}{2\operatorname{Var}[\cos(\omega^\mathsf{T}\Delta)] + \frac{2}{3}\frac{\varepsilon}{2}}\right) \\ &= 2T \exp\left(-\frac{D\varepsilon^2}{16\operatorname{Var}[\cos(\omega^\mathsf{T}\Delta)] + \frac{8}{3}\varepsilon}\right). \end{aligned} \tag{B.3}$$

This is a better bound when $\operatorname{Var}[\cos(\omega^\mathsf{T}\Delta)] + \frac{1}{6}\varepsilon < 1$. For pixie kernels, $\operatorname{Var}[\cos(\omega^\mathsf{T}\Delta)] \leq \frac{1}{2}$, so the Bernstein bound is better for any $\varepsilon < 3$. Since the maximal possible error is $\varepsilon = 2$, it is essentially always better for pixie kernels.

To unify the two bounds, let $\alpha_\varepsilon := \min\left(1, \max_{\Delta \in \mathcal{X}_\Delta} \frac{1}{2} + \frac{1}{2}k(2\Delta) - k(\Delta^2) + \frac{1}{6}\varepsilon\right)$. Then

$$\Pr\left(\bigcup_{i=1}^{T} |\tilde{f}(\Delta_i)| \geq \tfrac{1}{2}\varepsilon\right) \leq 2T \exp\left(-\frac{D\varepsilon^2}{16\alpha_\varepsilon}\right).$$

B.3.4 Optimizing Over r

Combining these two bounds, we have a bound in terms of r:

$$\Pr\left(\sup_{\Delta \in \mathcal{X}_\Delta} \left|\tilde{f}(\Delta)\right| \leq \varepsilon\right) \geq 1 - \kappa_1 r^{-d} - \kappa_2 r^2,$$

letting $\kappa_1 = 2(4\ell)^d \exp\left(-\frac{D\varepsilon^2}{16\alpha_\varepsilon}\right)$, $\kappa_2 = 4\sigma_p^2\varepsilon^{-2}$.

If we choose $r = (\kappa_1/\kappa_2)^{1/(d+2)}$, as did Rahimi and Recht (2007), the bound again becomes $1 - 2\kappa_1^{\frac{2}{d+2}}\kappa_2^{\frac{d}{d+2}}$. But we could instead maximize the bound by choosing r such that $d\kappa_1 r^{-d-1} - 2\kappa_2 r = 0$, i.e. $r = \left(\frac{d\kappa_1}{2\kappa_2}\right)^{\frac{1}{d+2}}$. Then the bound becomes $1 - \left(\left(\frac{d}{2}\right)^{\frac{-d}{d+2}} + \left(\frac{d}{2}\right)^{\frac{2}{d+2}}\right)\kappa_1^{\frac{2}{d+2}}\kappa_2^{\frac{d}{d+2}}$:

$$\begin{aligned} \Pr\left(\sup_{\Delta \in \mathcal{X}_\Delta} \left|\tilde{f}(\Delta)\right| > \varepsilon\right) &\leq \left(\left(\frac{d}{2}\right)^{\frac{-d}{d+2}} + \left(\frac{d}{2}\right)^{\frac{2}{d+2}}\right)\left(2(4\ell)^d \exp\left(-\frac{D\varepsilon^2}{16\alpha_\varepsilon}\right)\right)^{\frac{2}{d+2}} \left(4\sigma_p^2\varepsilon^{-2}\right)^{\frac{d}{d+2}} \\ &= \left(\left(\frac{d}{2}\right)^{\frac{-d}{d+2}} + \left(\frac{d}{2}\right)^{\frac{2}{d+2}}\right) 2^{\frac{2+4d+2d}{d+2}} \left(\frac{\sigma_p \ell}{\varepsilon}\right)^{\frac{2d}{d+2}} \exp\left(-\frac{D\varepsilon^2}{8(d+2)\alpha_\varepsilon}\right) \end{aligned} \tag{B.4}$$

$$= \left(\left(\frac{d}{2}\right)^{\frac{-d}{d+2}} + \left(\frac{d}{2}\right)^{\frac{2}{d+2}} \right) 2^{\frac{6d+2}{d+2}} \left(\frac{\sigma_p \ell}{\varepsilon}\right)^{\frac{2}{1+2/d}} \exp\left(-\frac{D\varepsilon^2}{8(d+2)\alpha_\varepsilon}\right). \tag{B.5}$$

For $\varepsilon \le \sigma_p \ell$, we can loosen the exponent on the middle term to 2, though in low dimensions we have a somewhat sharper bound. We no longer need the $\ell > 1$ assumption of the original proof.

To prove the final statement of Proposition 3.6, simply set (B.4) to be at most δ and solve for D.

B.4 Proof of Proposition 3.7

We will follow the proof strategy of Proposition 3.6 as closely as possible.

Our approximation is now $\breve{s}(x, y) = \breve{z}(x)^\mathsf{T} \breve{z}(y)$, and the error is $\breve{f}(x, y) = \breve{s}(x, y) - k(y, x)$. Note that $\breve{s}$ and $\breve{f}$ are not shift-invariant: for example, with $D = 1$, $\breve{s}(x, y) = \cos(\omega^\mathsf{T}\Delta) + \cos(\omega^\mathsf{T}(x + y) + 2b)$ but $\breve{s}(\Delta, 0) = \cos(\omega^\mathsf{T}\Delta) + \cos(\omega^\mathsf{T}\Delta + 2b)$.

Let $q = \begin{bmatrix} x \\ y \end{bmatrix} \in \mathcal{X}^2$ denote the argument to these functions. $\mathcal{X}^2$ is a compact set in $\mathbb{R}^{2d}$ with diameter $\sqrt{2}\ell$, so we can cover it with an ε-net using at most $T = \left(2\sqrt{2}\ell/r\right)^{2d}$ balls of radius r. Let $\{q_i\}_{i=1}^T$ denote their centers, and L_f be the Lipschitz constant of $f : \mathbb{R}^{2d} \to \mathbb{R}$.

B.4.1 Regularity Condition

To show $\mathbb{E}\, \nabla \breve{s}(q) = \nabla\, \mathbb{E}\, \breve{s}(q)$, we can define $\breve{g}^i_{x,y}(t, \omega)$ analogously to in Appendix B.3.1, where here ω contains all the ω_i and b_i variables used in $\breve{z}$. We then have:

$$\begin{aligned}
\mathbb{E}_\omega \left| \frac{\partial \breve{g}^i_{x,y}(t, \omega)}{\partial t} \right| &= \mathbb{E}_\omega \left| \frac{1}{D} \sum_{j=1}^{D} -\omega_{ji} \cos(\omega_j^\mathsf{T} y + b_j) \sin(\omega_j^\mathsf{T} x + t\omega_{ji} + b_j) \right| \\
&\le \mathbb{E}_\omega \left[\frac{1}{D} \sum_{j=1}^{D} \left|\omega_{ji}\right| \right] \le \mathbb{E}_\omega \left|\omega\right|,
\end{aligned}$$

which we have assumed to be finite.

B.4.2 Lipschitz Constant

The argument follows that of Appendix B.3.2 up to (B.2), using q^* in place of Δ^*. Then:

$$\begin{aligned}
\mathbb{E}[L_{\breve{f}}^2] &\le \mathbb{E}\, \|\nabla \breve{s}(q^*)\|^2 \\
&= \mathbb{E} \left\| \nabla_q \left(2\cos(\omega^\mathsf{T} x + b) \cos(\omega^\mathsf{T} y + b) \right) \right\|^2 \\
&= \mathbb{E} \left[\left\| \nabla_x \left(2\cos(\omega^\mathsf{T} x + b) \cos(\omega^\mathsf{T} y + b) \right) \right\|^2 + \left\| \nabla_y \left(2\cos(\omega^\mathsf{T} x + b) \cos(\omega^\mathsf{T} y + b) \right) \right\|^2 \right]
\end{aligned}$$

$$
\begin{aligned}
&= \mathbb{E}\left[\left\|-2\sin(\omega^\mathsf{T} x^* + b)\cos(\omega^\mathsf{T} y^* + b)\,\omega\right\|^2 + \left\|-2\cos(\omega^\mathsf{T} x^* + b)\sin(\omega^\mathsf{T} y^* + b)\,\omega\right\|^2\right] \\
&= \mathbb{E}\left[4\left(\sin^2(\omega^\mathsf{T} x^* + b)\cos^2(\omega^\mathsf{T} y^* + b) + \cos^2(\omega^\mathsf{T} x^* + b)\sin^2(\omega^\mathsf{T} y^* + b)\right)\|\omega\|^2\right] \\
&= \mathbb{E}_\omega\left[\mathbb{E}_b\left[2 - \cos(2\omega^\mathsf{T}(x^* - y^*)) - \cos(2\omega^\mathsf{T}(x^* + y^*) + 4b)\right]\|\omega\|^2\right] \\
&= \mathbb{E}_\omega\left[\left(2 - \cos(2\omega^\mathsf{T}(x^* - y^*))\right)\|\omega\|^2\right] \\
&\leq 3\,\mathbb{E}\,\|\omega\|^2 = 3\sigma_p^2.
\end{aligned}
$$

Following through with Markov's inequality:

$$
\Pr\left(L_{\breve{f}} \geq \varepsilon/(2r)\right) \leq 3\sigma_p^2(2r/\varepsilon)^2 = 12(\sigma_p r/\varepsilon)^2.
$$

B.4.3 Anchor Points

For any fixed x, y, $\breve{s}$ takes a mean of D terms with expectation $k(x, y)$ bounded by ±2. Using Hoeffding's inequality:

$$
\Pr\left(\bigcup_{i=1}^{T} |\breve{f}(q_i)| \geq \tfrac{1}{2}\varepsilon\right) \leq T\Pr\left(|\breve{f}(q)| \geq \tfrac{1}{2}\varepsilon\right) \leq 2T\exp\left(-\frac{2D\left(\frac{\varepsilon}{2}\right)^2}{(2-(-2))^2}\right) = 2T\exp\left(-\frac{D\varepsilon^2}{32}\right).
$$

Since the variance of each term is given by (3.6) as $\mathrm{Var}[\cos(\omega^\mathsf{T}\Delta)] + \frac{1}{2}$, we can instead use Bernstein's inequality:

$$
\begin{aligned}
T\Pr\left(\left|\breve{f}(\Delta)\right| > \tfrac{1}{2}\varepsilon\right) &\leq 2T\exp\left(-\frac{D\frac{\varepsilon^2}{4}}{2\left(\mathrm{Var}[\cos(\omega^\mathsf{T}\Delta)] + \frac{1}{2}\right) + \frac{2}{3}2\frac{\varepsilon}{2}}\right) \\
&= 2T\exp\left(-\frac{D\varepsilon^2}{4 + 8\,\mathrm{Var}[\cos(\omega^\mathsf{T}\Delta)] + \frac{8}{3}\varepsilon}\right).
\end{aligned}
\tag{B.6}
$$

Thus Bernstein's gives us a tighter bound if

$$
4 + 8\,\mathrm{Var}[\cos(\omega^\mathsf{T}\Delta)] + \frac{8}{3}\varepsilon < 32 \qquad \text{i.e.} \qquad 2\,\mathrm{Var}[\cos(\omega^\mathsf{T}\Delta)] + \frac{2}{3}\varepsilon < 7,
$$

which since $\mathrm{Var}[\cos(\omega^\mathsf{T}\Delta)] \leq 1$, means the Bernstein bound is better for any $\varepsilon < 7.5$ no matter the kernel.

Still, it can be preferable to have a bound independent of ε, so to unify the bounds define $\alpha'_\varepsilon = \min\left(1, \max_\Delta \frac{1}{8} + \frac{1}{4}\mathrm{Var}[\cos(\omega^\mathsf{T}\Delta)] + \frac{1}{12}\varepsilon\right)$; then

$$
\Pr\left(\bigcup_{i=1}^{T} |\breve{f}(q_i)| \geq \tfrac{1}{2}\varepsilon\right) \leq 2T\exp\left(-\frac{D\varepsilon^2}{32\alpha'_\varepsilon}\right).
$$

B.4.4 Optimizing Over r

Our bound is now of the form

$$\Pr\left(\sup_{q\in\mathcal{X}^2}\left|\breve{f}(q)\right| \le \varepsilon\right) \ge 1 - \kappa_1 r^{-2d} - \kappa_2 r^2,$$

with $\kappa_1 = 2\left(2\sqrt{2}\ell\right)^{2d}\exp\left(-\frac{D\varepsilon^2}{32\alpha'_\varepsilon}\right)$ and $\kappa_2 = 12\sigma_p^2\varepsilon^{-2}$.

This is maximized by r when $2d\kappa_1 r^{-2d-1} - 2\kappa_2 r = 0$, i.e. $r = \left(\frac{d\kappa_1}{\kappa_2}\right)^{\frac{1}{2d+2}}$. Substituting that value of r into the bound yields $1 - \left(d^{\frac{-d}{d+1}} + d^{\frac{1}{d+1}}\right)\kappa_1^{\frac{1}{d+1}}\kappa_2^{\frac{d}{d+1}}$, and thus:

$$\begin{aligned}\Pr\left(\sup_{q\in\mathcal{X}^2}\left|\breve{f}(q)\right| > \varepsilon\right) &\le \left(d^{\frac{-d}{d+1}} + d^{\frac{1}{d+1}}\right)\left(2\left(2\sqrt{2}\ell\right)^{2d}\exp\left(-\frac{D\varepsilon^2}{32\alpha'_\varepsilon}\right)\right)^{\frac{1}{d+1}}\left(12\sigma_p^2\varepsilon^{-2}\right)^{\frac{d}{d+1}} \\ &= \left(d^{\frac{-d}{d+1}} + d^{\frac{1}{d+1}}\right)2^{\frac{1+2d+d+2d}{d+1}}3^{\frac{d}{d+1}}\left(\frac{\sigma_p\ell}{\varepsilon}\right)^{\frac{2d}{d+1}}\exp\left(-\frac{D\varepsilon^2}{32(d+1)\alpha'_\varepsilon}\right) \qquad \text{(B.7)} \\ &= \left(d^{\frac{-d}{d+1}} + d^{\frac{1}{d+1}}\right)2^{\frac{5d+1}{d+1}}3^{\frac{d}{d+1}}\left(\frac{\sigma_p\ell}{\varepsilon}\right)^{\frac{2}{1+1/d}}\exp\left(-\frac{D\varepsilon^2}{32(d+1)\alpha'_\varepsilon}\right).\end{aligned}$$

As before, when $\varepsilon \le \sigma_p\ell$ we can loosen the exponent on the middle term to 2; it is slightly worse than the corresponding exponent of (B.5) for small d.

To prove the final statement of Proposition 3.7, set (B.7) to be at most δ and solve for D.

B.5 Proof of Proposition 3.8

Consider the $\tilde{z}$ features, and recall that we supposed k is L-Lipschitz over $\mathcal{X}_\Delta := \{x - y \mid x, y \in \mathcal{X}\}$.

Our primary tool will be the following slight generalization of Dudley's entropy integral, which is a special case of Lemma 13.1 of Boucheron et al. (2013). (The only difference from their Corollary 13.2 is that we maintain the variance factor v.)

Theorem (Boucheron et al. 2013)**.** *Let $\mathcal{T}$ be a finite pseudometric space and let $(X_t)_{t\in\mathcal{T}}$ be a collection of random variables such that for some constant $v > 0$,*

$$\log\mathbb{E}\,e^{\lambda(X_t - X_{t'})} \le \frac{1}{2}v\lambda^2 d^2(t,t')$$

for all $t, t' \in \mathcal{T}$ and all $\lambda > 0$. For any $t_0 \in \mathcal{T}$, let $\delta = \sup_{t\in\mathcal{T}} d(t, t_0)$; then

$$\mathbb{E}\left[\sup_{t\in\mathcal{T}} X_t - X_{t_0}\right] \le 12\sqrt{v}\int_0^{\delta/2}\sqrt{H(u,\mathcal{T})}\,\mathrm{d}u$$

where $H(u,\mathcal{T})$ is the metric entropy *of $\mathcal{T}$, that is, the logarithm of the δ-packing number $N(u,\mathcal{T})$.*

Note that, although stated for finite pseudometric spaces, the result is extensible to seperable pseudometric spaces (such as $\mathcal{X}_\Delta$) by standard arguments.

The δ-packing number is at most the $\frac{\delta}{2}$-covering number, which Proposition 5 of Cucker and Smale (2001) bounds. Thus, picking $\Delta_0 = 0$ gives $\delta = \ell$, $H(\delta, \mathcal{X}_\Delta) \le d \log(8\ell/\delta)$, and

$$\int_0^{\ell/2} \sqrt{H(u, \mathcal{X}_\Delta)}\, \mathrm{d}u \le \int_0^{\ell/2} \sqrt{d \log(8\ell/u)}\, \mathrm{d}u = \gamma \ell \sqrt{d},$$

where $\gamma := 4\sqrt{\pi}\, \mathrm{erfc}(2\sqrt{\log 2}) + \sqrt{\log 2} \approx 0.964$.

Now, $\frac{2}{D}\left(\cos(\omega_i^\mathsf{T}\Delta) - k(\Delta) - \cos(\omega_i^\mathsf{T}\Delta') + k(\Delta')\right)$ has mean zero, and absolute value at most

$$\begin{aligned}
\left|\frac{2}{D}\left(\cos(\omega_i^\mathsf{T}\Delta) - k(\Delta) - \cos(\omega_i^\mathsf{T}\Delta') + k(\Delta')\right)\right| &\le \frac{2}{D}\left(\left|\cos(\omega_i^\mathsf{T}\Delta) - \cos(\omega_i^\mathsf{T}\Delta')\right| + |k(\Delta) - k(\Delta')|\right) \\
&\le \frac{2}{D}\left(\left|\omega_i^\mathsf{T}\Delta - \omega_i^\mathsf{T}\Delta'\right| + L\,\|\Delta - \Delta'\|\right) \\
&\le \frac{2}{D}\left(\|\omega_i\| + L\right)\|\Delta - \Delta'\|. \qquad \text{(B.8)}
\end{aligned}$$

Thus, via Hoeffding's lemma (Boucheron et al. 2013, Lemma 2.2), each such term has log moment generating function at most $\frac{2}{D^2}(\|\omega_i\| + L)^2 \lambda^2 \|\Delta - \Delta'\|^2$.

This is almost in the form required by Dudley's entropy integral, except that ω_i is a random variable. Thus, for any $r > 0$, define the random process $\tilde{g}_r$ which is distributed as $\tilde{f}$ except we require that $\|\omega_1\| = r$ and $\|\omega_i\| \le r$ for all $i > 1$. Since log mgfs of independent variables are additive, we thus have

$$\log \mathbb{E}\, e^{\lambda(\tilde{g}_r(\Delta) - \tilde{g}_r(\Delta'))} \le \frac{1}{D}\left(\frac{2}{D}\sum_{i=1}^{D/2}(\|\omega_i\| + L)^2\right)\lambda^2\|\Delta - \Delta'\|^2 \le \frac{1}{D}(r + L)^2\lambda^2\|\Delta - \Delta'\|^2.$$

$\tilde{g}_r$ satisfies the conditions of the theorem with $v = \frac{1}{D}(r + L)^2$. Now, $\tilde{g}_r(0) = 0$, so we have

$$\mathbb{E}\left[\sup_{\Delta \in \mathcal{X}_\Delta} \tilde{g}_r(\Delta)\right] \le \frac{12\gamma\sqrt{d}\ell}{\sqrt{D}}(r + L).$$

But the distribution of $\tilde{f}$ conditioned on the event $\max_i \|\omega_i\| = r$ is the same as the distribution of $\tilde{g}_r$. Thus

$$\mathbb{E} \sup \tilde{f} = \mathbb{E}_r\left[\mathbb{E}[\sup \tilde{g}_r]\right] \le \mathbb{E}_r\left[\frac{12\gamma\sqrt{d}\ell}{\sqrt{D}}(r + L)\right] = \frac{12\gamma\sqrt{d}\ell}{\sqrt{D}}(R + L)$$

where $R := \mathbb{E} \max_{i=1}^{D/2} \|\omega_i\|$.

The same holds for $\mathbb{E} \sup(-\tilde{f})$. Since we have $\sup \tilde{f} \ge 0$, $\sup(-\tilde{f}) \ge 0$, the claim follows from $\mathbb{E}\left[\max(\sup \tilde{f}, \sup(-\tilde{f}))\right] \le \mathbb{E}\left[\sup \tilde{f} + \sup(-\tilde{f})\right]$.

B.6 Proof of Proposition 3.9

For the $\breve{z}$ features,the error process again must be defined over $\mathcal{X}^2$ due to the non-shift invariant noise. We still assume that k is L-Lipschitz over $\mathcal{X}_\Delta$, however.

Compared to the argument of Appendix B.5, we have $H(u, \mathcal{X}^2) \le 2d \log\left(4\sqrt{2}\ell/u\right)$. Unlike $\mathcal{X}_\Delta$, however, $\mathcal{X}^2$ does not necessarily contain an obvious point q_0 to minimize $\sup_{q\in\mathcal{X}^2} d(q, q_0)$, nor an obvious minimal value. We rather consider the "radius" $\rho := \sup_{x\in\mathcal{X}} d(x, x_0)$, achieved by any convenient point x_0; then $\sup_{q\in\mathcal{X}^2} d\left(q, (x_0, x_0)\right) = \sqrt{2}\rho$. Note that $\frac{1}{2}\ell \le \rho \le \ell$, where the lower bound is achieved by $\mathcal{X}$ a ball, and the upper bound by $\mathcal{X}$ a sphere. The integral in the bound is then

$$\begin{aligned}\int_0^{\rho/\sqrt{2}} \sqrt{H(u, \mathcal{X}^2)} &\le \int_0^{\rho/\sqrt{2}} \sqrt{2d\log(4\sqrt{2}\ell/u)} \\ &= 4\sqrt{\pi d}\ell \operatorname{erfc}\left(\sqrt{\tfrac{1}{2}\log 2 + \log 4\sqrt{2}\tfrac{\ell}{\rho}}\right) + \rho\sqrt{d}\sqrt{\tfrac{5}{2}\log 2 + \log\tfrac{\ell}{\rho}} \\ &= \left(4\sqrt{\pi}\operatorname{erfc}\left(\sqrt{\tfrac{1}{2}\log 2 + \log 4\sqrt{2}\tfrac{\ell}{\rho}}\right) + \tfrac{\rho}{\ell}\sqrt{\tfrac{5}{2}\log 2 + \log\tfrac{\ell}{\rho}}\right)\ell\sqrt{d}. \qquad \text{(B.9)}\end{aligned}$$

Calling the term in parentheses $\gamma'_{\ell/\rho}$, we have that $\gamma'_1 \approx 1.541$, $\gamma'_2 \approx 0.803$, and it decreases monotonically in between, as shown in Figure B.1.

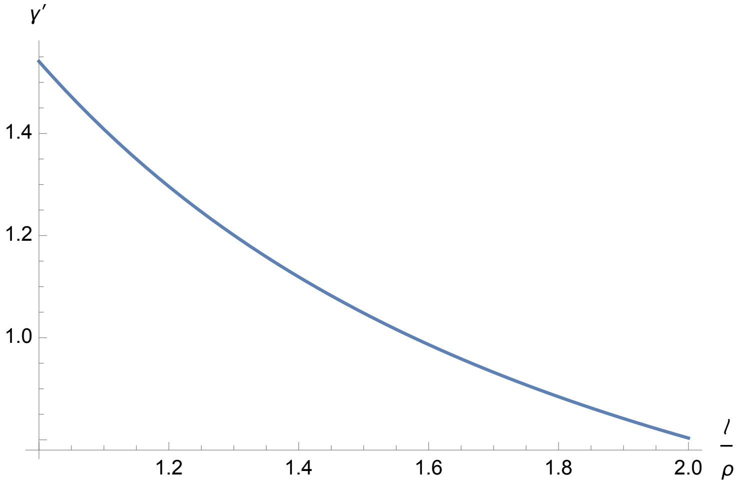


Figure B.1: The coefficient of (B.9) as a function of ℓ/ρ.

We will again use the notation of $q = (x, y) \in \mathcal{X}^2$, $\Delta = x - y$, $t = x + y$. Each term in the sum of $\breve{f}(q) - \breve{f}(q')$ has mean zero and absolute value at most

$$\begin{aligned}&\frac{1}{D}\left|\cos(\omega_i^\mathsf{T}\Delta) + \cos(\omega_i^\mathsf{T} t + 2b_i) - k(\Delta) - \cos(\omega_i^\mathsf{T}\Delta') + \cos(\omega_i^\mathsf{T} t' + 2b_i) + k(\Delta')\right| \\ &\qquad \le \frac{1}{D}\left(\left|\cos(\omega_i^\mathsf{T}\Delta) - \cos(\omega_i^\mathsf{T}\Delta')\right| + \left|\cos(\omega_i^\mathsf{T} t + 2b_i) - \cos(\omega_i^\mathsf{T} t' + 2b_i)\right| + \left|k(\Delta) - k(\Delta')\right|\right)\end{aligned}$$

$$\leq \frac{1}{D}\left(\|\omega_i\|\|\Delta - \Delta'\| + \|\omega_i\|\|t - t'\| + L\|\Delta - \Delta'\|\right).$$

Now, in order to cast this in terms of distance on $\mathcal{X}^2$, let $\delta_x = x - x'$, $\delta_y = y - y'$. Then

$$\begin{aligned}
\|q - q'\|^2 &= \|\delta_x\|^2 + \|\delta_y\|^2 \\
(\|\Delta - \Delta'\| + \|t - t'\|)^2 &= \left(\sqrt{\|\delta_x\|^2 + \|\delta_y\|^2 - 2\delta_x^\mathsf{T}\delta_y} + \sqrt{\|\delta_x\|^2 + \|\delta_y\|^2 + 2\delta_x^\mathsf{T}\delta_y}\right)^2 \\
&= 2\|\delta_x\|^2 + 2\|\delta_y\|^2 + 2\sqrt{\left(\|\delta_x\|^2 + \|\delta_y\|^2\right)^2 - 4(\delta_x^\mathsf{T}\delta_y)^2} \\
&\leq 4\left(\|\delta_x\|^2 + \|\delta_y\|^2\right) \\
\|\Delta - \Delta'\| + \|t - t'\| &\leq 2\|q - q'\| \\
\|\Delta - \Delta'\| &\leq 2\|q - q'\|
\end{aligned}$$

and so each term in the sum of $\breve{f}(q) - \breve{f}(q')$ has absolute value at most $\frac{2}{D}\left(\|\omega_i\| + L\right)\|q - q'\|$. Note that this agrees exactly with (B.8), but the sum in $\breve{f}(q) - \breve{f}(q')$ has D terms rather than $D/2$. Defining $\breve{g}_r$ analogously to $\tilde{g}_r$, we thus get that

$$\log \mathbb{E}\, e^{\lambda(\breve{g}_r(q) - \breve{g}_r(q'))} \leq \frac{2}{D}\left(\frac{1}{D}\sum_{i=1}^{D}(\|\omega_i\| + L)^2\right)\lambda^2\|q - q'\|^2 \leq \frac{2}{D}(r + L)^2\lambda^2\|q - q'\|^2,$$

and the conditions of the theorem hold with $v = \frac{4}{D}(r + L)^2$. Note that $\mathbb{E}\,\breve{g}_r(q_0) = 0$. Carrying out the rest of the argument, we get that

$$\mathbb{E}\sup \breve{f} = \mathbb{E}_r[\mathbb{E}[\sup \breve{g}_r]] \leq \mathbb{E}_r\left[\frac{24\beta_{\ell/\rho}\ell\sqrt{d}}{\sqrt{D}}(r + L)\right] = \frac{24\beta_{\ell/\rho}\ell\sqrt{d}}{\sqrt{D}}(R + L),$$

and similarly for $\mathbb{E}\sup \breve{f}$. We do not have a guarantee that $\breve{f}(q)$ does not have a consistent sign, and so our bound becomes

$$\begin{aligned}
\mathbb{E}\|\breve{f}\|_\infty &\leq \mathbb{E}\left[\|\breve{f}\|_\infty \mid \breve{f} \text{ crosses } 0\right] \Pr\left(\breve{f} \text{ crosses } 0\right) + 3\Pr\left(\breve{f} \text{ does not cross } 0\right) \\
&\leq \frac{48\beta_{\ell/\rho}\ell\sqrt{d}}{\sqrt{D}}(R' + L)\Pr\left(\breve{f} \text{ crosses } 0\right) + 3\Pr\left(\breve{f} \text{ does not cross } 0\right).
\end{aligned}$$

$\Pr\left(\breve{f} \text{ crosses } 0\right)$ is extremely close to 1 in "usual" situations.

Appendix C

Proofs for Chapter 4

C.1 Proof of Proposition 4.10

We will now prove the bound on the error probability of our embedding

$$\Pr\left(\left|K(p,q) - z(\hat{A}(\hat{p}))^\mathsf{T} z(\hat{A}(\hat{q}))\right| \geq \varepsilon\right)$$

for fixed densities p and q.

Setup We will need a few assumptions on the densities:

1. p and q are bounded above and below: for $x \in [0,1]^d$, $0 < \rho_* \leq p(x), q(x) \leq \rho^* < \infty$.
2. $p, q \in \Sigma(\beta, L_\beta)$ for some $\beta, L_\beta > 0$. $\Sigma(\beta, L)$ refers to the Hölder class of functions f whose partial derivatives up to order $\lfloor \beta \rfloor$ are continuous and whose rth partial derivatives, where r is a multi-index of order $\lfloor \beta \rfloor$, satisfy $|D^r f(x) - D^r f(y)| \leq L\|x - y\|^\beta$. Here $\lfloor \beta \rfloor$ is the greatest integer *strictly* less than β.
3. p, q are periodic.

These are fairly standard smoothness assumptions in the nonparametric estimation literature.

Let $\gamma = \min(\beta, 1)$. If $\beta > 1$, then $p, q \in \Sigma(1, L_\gamma)$ for some L_γ; otherwise, clearly $p, q \in \Sigma(\beta, L_\beta)$. Then, from assumption 3, $p, q \in \Sigma_{\text{per}}(\gamma, L_\gamma)$, the periodic Hölder class. We'll need this to establish the Sobolev ellipsoid containing p and q.

We will use kernel density estimation with a bounded, continuous kernel so that the bound of Giné and Guillou (2002) applies, with bandwidth $h \asymp n^{-1/(2\beta+d)} \log n$, and truncating density estimates to $[\rho_*, \rho^*]$.

We also use the Fourier basis $\varphi_\alpha = \exp\left(2\mathbb{i}\pi\alpha^\mathsf{T} x\right)$, and define V as the set of indices α s.t. $\sum_{j=1}^d |\alpha_j|^{2s} \leq t$ for parameters $0 < s \leq 1$, $t > 0$ to be discussed later.

Decomposition Let $r_\sigma(\Delta) = \exp\left(-\Delta^2/(2\sigma^2)\right)$. Then

$$\begin{aligned}\left|K(p,q) - z(\hat{A}(\hat{p}))^\mathsf{T} z(\hat{A}(\hat{q}))\right| &\leq \\ &\left|K(p,q) - r_{\sigma_k}\left(\|\hat{A}(\hat{p}) - \hat{A}(\hat{q})\|\right)\right| + \left|r_{\sigma_k}\left(\|\hat{A}(\hat{p}) - \hat{A}(\hat{q})\|\right) - z(\hat{A}(\hat{p}))^\mathsf{T} z(\hat{A}(\hat{q}))\right|.\end{aligned}$$

The latter term was bounded by Chapter 3. For the former, note that r_σ is $\frac{1}{\sigma\sqrt{e}}$-Lipschitz, so the first term is at most $\frac{1}{\sigma_k\sqrt{e}}\left|\rho(p,q) - \|\hat{A}(\hat{p}) - \hat{A}(\hat{q})\|\right|$. Breaking this up with the triangle inequality:

$$\begin{aligned}\left|\rho(p,q) - \|\hat{A}(\hat{p}) - \hat{A}(\hat{q})\|\right| &\le |\rho(p,q) - \rho(\hat{p},\hat{q})| + |\rho(\hat{p},\hat{q}) - \|\psi(\hat{p}) - \psi(\hat{q})\|| \\ &+ |\|\psi(\hat{p}) - \psi(\hat{q})\| - \|A(\hat{p}) - A(\hat{q})\|| + \left|\|A(\hat{p}) - A(\hat{q})\| - \|\hat{A}(\hat{p}) - \hat{A}(\hat{q})\|\right|. \quad \text{(C.1)}\end{aligned}$$

Estimation error Recall that ρ is a metric, so the reverse triangle inequality allows us to address the first term with

$$|\rho(p,q) - \rho(\hat{p},\hat{q})| \le \rho(p,\hat{p}) + \rho(q,\hat{q}).$$

For ρ^2 the total variation, squared Hellinger, or Jensen-Shannon HDDs, we have that $\rho^2(p,\hat{q}) \le \textsc{tv}(p,\hat{p})$ (J. Lin 1991). Moreover, as the distributions are supported on $[0,1]^d$, $\textsc{tv}(p,\hat{p}) = \frac{1}{2}\|p-\hat{p}\|_1 \le \frac{1}{2}\|p-\hat{p}\|_\infty$.

It is a consequence of Giné and Guillou (2002) that, for any $\delta > 0$,

$$\Pr\left(\|p-\hat{p}\|_\infty > \frac{\sqrt{C_\delta \log n}}{n^{\beta/(2\beta+d)}}\right) < \delta$$

for some C_δ depending on the kernel. Thus

$$\Pr\left(|\rho(p,q) - \rho(\hat{p},\hat{q})| \ge \varepsilon\right) < 2C^{-1}\left(\frac{\varepsilon^4 n^{2\beta/(2\beta+d)}}{4\log n}\right)$$

, where $C_{C^{-1}(x)} = x$.

λ approximation The second term of (C.1), the approximation error due to sampling λs, admits a simple Hoeffding bound. Note that $\left\|\hat{p}^R_\lambda - \hat{q}^R_\lambda\right\|^2 + \left\|\hat{p}^I_\lambda - \hat{q}^I_\lambda\right\|^2$, viewed as a random variable in λ only, has expectation $\rho^2(\hat{p},\hat{q})$ and is bounded by $[0,4Z]$ (where $Z = \int_{\mathbb{R}_{\ge 0}} \mathrm{d}\mu(\lambda)$): write it as $Z\int|\hat{p}(x)^{\frac{1}{2}+\mathbb{i}\lambda} - \hat{q}(x)^{\frac{1}{2}+\mathbb{i}\lambda}|^2\,\mathrm{d}x$, expand the square, and use $\int\sqrt{\hat{p}(x)\hat{q}(x)}\mathrm{d}x \le 1$ (via Cauchy-Schwarz).

For nonnegative random variables X and Y, $\Pr\left(|X-Y| \ge \varepsilon\right) \le \Pr\left(\left|X^2 - Y^2\right| \ge \varepsilon^2\right)$, so we have that $\Pr\left(|\|\psi(\hat{p}) - \psi(\hat{q})\| - \rho(\hat{p},\hat{q})| \ge \varepsilon\right)$ is at most $2\exp(-M\varepsilon^4/(8Z^2))$.

Tail truncation error The third term of (C.1), the error due to truncating the tail projection coefficients of the p^S_λ functions, requires a little more machinery. First note that $\left|\|\psi(\hat{p}) - \psi(\hat{q})\|^2 - \|A(\hat{p}) - A(\hat{q})\|^2\right|$ is at most

$$\sum_{j=1}^M \sum_{S=R,I} \sum_{\alpha\notin V} \left|a_\alpha(\hat{p}^S_\lambda - \hat{q}^S_\lambda)\right|^2.$$

Let $\mathcal{W}(s,L)$ be the Sobolev ellipsoid of functions $\sum_{\alpha\in\mathbb{Z}^d} a_\alpha\varphi_\alpha$ such that $\sum_{\alpha\in\mathbb{Z}^d}\left(\sum_{j=1}^d|\alpha_j|^{2s}\right)|a_\alpha|^2 \le L$, where φ is still the Fourier basis. Then Lemma 14 of Krishnamurthy et al. (2014) shows that $\Sigma_{\mathrm{per}}(\gamma, L_\gamma) \subseteq \mathcal{W}(s,L')$ for any $0 < s < \gamma$ and $L' = dL_\gamma^2(2\pi)^{-2\lfloor\gamma\rfloor}\frac{4^\gamma}{4^\gamma - 4^s}$.

So, suppose that $\hat{p}, \hat{q} \in \Sigma_{\text{per}}(\hat{\gamma}, \widehat{L})$ with probability at least $1 - \delta$. Since $x \mapsto x^{\frac{1}{2}+\mathbb{i}\lambda}$ is $\frac{\sqrt{1+4\lambda^2}}{2\sqrt{\rho_*}}$-Lipschitz on $[\rho_*, \infty)$, $\hat{p}^S_\lambda \in \Sigma_{\text{per}}\left(\hat{\gamma}, \frac{1}{2}\sqrt{1+4\lambda^2}\,\widehat{L}\,\rho_*^{-\frac{1}{2}}\right)$ and so $\hat{p}^S_\lambda - \hat{q}^S_\lambda$ is in $\mathcal{W}(s, (1+4\lambda^2)\widehat{L}')$ for $s < \hat{\gamma}$ and $\widehat{L}' = d\widehat{L}^2\rho_*^{-1}/(1-4^{s-\hat{\gamma}})$.

Recall that we chose V to be the set of $\alpha \in \mathbb{Z}^d$ such that $\sum_{j=1}^d |\alpha_j|^{2s} \le t$. Thus

$$\begin{aligned}\sum_{\alpha \notin V} |a_\alpha(\hat{p}^S_\lambda - \hat{q}^S_\lambda)|^2 &\le \sum_{\alpha \notin V} |a_\alpha(\hat{p}^S_\lambda - \hat{q}^S_\lambda)|^2 \left(\sum_{j=1}^d |\alpha_j|^{2s}\right)/t \\ &\le (1+4\lambda^2)\widehat{L}'/t.\end{aligned}$$

The tail error term is therefore at least ε with probability no more than

$$\delta + 2\sum_{j=1}^M \Pr\left((1+4\lambda_j^2)\widehat{L}'/t \ge \varepsilon^2/(2M)\right).$$

The latter probability, of course, depends on the choice of HDD ρ. Letting $\zeta = t\varepsilon^2/(8M\widehat{L}') - \frac{1}{4}$, it is 1 if $\zeta < 0$ and $1 - \mu\left([0, \sqrt{\zeta}]\right)/Z$ otherwise. If $\zeta \ge 0$, squared Hellinger's probability is 0, and total variation's is $\frac{2}{\pi}\arctan(\sqrt{\zeta})$. A closed form for the cumulative distribution function for the Jensen-Shannon measure is unfortunately unknown.

Numerical integration error The final term of (C.1) also bears a Hoeffding bound. Define the projection coefficient difference $\Delta^S_{\lambda,\alpha}(p,q) = a_{\alpha,\lambda}(p^S_\lambda) - a_\alpha(q^S_\lambda)$, and $\hat{\Delta}$ similarly but with $\hat{a}$. Then

$$\left| \|A(\hat{p}) - A(\hat{q})\|^2 - \left\|\hat{A}(\hat{p}) - \hat{A}(\hat{q})\right\|^2 \right| \le \sum_{j=1}^M \sum_{S=R,I} \sum_{\alpha \in V} \left| \left|\Delta^S_{\alpha,\lambda_j}(\hat{p},\hat{q})\right|^2 - \left|\hat{\Delta}^S_{\alpha,\lambda_j}(\hat{p},\hat{q})\right|^2 \right|. \qquad \text{(C.2)}$$

Letting $\hat{\epsilon}(p) = a_\alpha(\hat{p}^S_\lambda) - \hat{a}_\alpha(\hat{p}^S_\lambda)$, each summand is at most $(\hat{\epsilon}(p)+\hat{\epsilon}(q))^2 + 2\left|\Delta^S_{\lambda,\alpha}(\hat{p},\hat{q})\right|(\hat{\epsilon}(p)+\hat{\epsilon}(q))$. Also, $\left|\Delta^S_{\alpha,\lambda}(\hat{p},\hat{q})\right| \le 2\sqrt{Z}$, using Cauchy-Schwarz on the integral and $\|\varphi_\alpha\|_2 = 1$. Thus each summand in (C.2) can be more than ε only if one of the $\hat{\epsilon}$s is more than $\sqrt{Z + \varepsilon/4} - \sqrt{Z}$.

Now, using (4.10), $\hat{a}_\alpha(\hat{p}^S_\lambda)$ is an empirical mean of n_e independent terms, each with absolute value bounded by $(\sqrt{\rho^*}+1)\max_x|\varphi_\alpha(x)| = \sqrt{\rho^*}+1$. Thus, using a Hoeffding bound on the $\hat{\epsilon}$s, we get that $\Pr\left(\left|\|A(\hat{p}) - A(\hat{q})\|^2 - \|\hat{A}(\hat{p}) - \hat{A}(\hat{q})\|^2\right| \ge \varepsilon\right)$ is no more than $8MS\exp\left(-\frac{n_e\left(\sqrt{Z+\varepsilon^2/(8S)}-\sqrt{Z}\right)^2}{2Z(\sqrt{\rho^*}+1)^2}\right)$.

Final bound Combining the bounds for the decomposition (C.1) with the pointwise rate for RKS features, we get:

$$\Pr\left(\left|K(p,q) - z(\hat{A}(\hat{p})^\mathsf{T} z(\hat{A}(\hat{q}))\right| \ge \varepsilon\right) \le$$

$$
\begin{aligned}
2\exp\left(-D\varepsilon_{\mathrm{RKS}}^{2}\right) + 2C^{-1}\left(\frac{\varepsilon_{\mathrm{KDE}}^{4} n^{2\beta/(2\beta+d)}}{4\log n}\right) \\
+ 2\exp\left(-M\varepsilon_{\lambda}^{4}/(8Z^{2})\right) \\
+ \delta + 2M\left(1 - \mu\left[0, \sqrt{\max\left(0, \frac{\rho_{*} t \varepsilon_{\mathrm{tail}}^{2}}{8Md\widehat{L}^{2}} \frac{4^{\hat{\gamma}} - 4^{s}}{4^{\hat{\gamma}}} - \frac{1}{4}\right)}\right)\right) \\
+ 8M\left|V\right| \exp\left(-\tfrac{1}{2} n_{e}\left(\frac{\sqrt{1 + \varepsilon_{\mathrm{int}}^{2}/(8\left|V\right| Z)} - 1}{\sqrt{\rho^{*}} + 1}\right)^{2}\right)
\end{aligned}
$$

for any $\varepsilon_{\mathrm{RKS}} + \frac{1}{\sigma_k \sqrt{e}} \left(\varepsilon_{\mathrm{KDE}} + \varepsilon_{\lambda} + \varepsilon_{\mathrm{tail}} + \varepsilon_{\mathrm{int}}\right) \leq \varepsilon$.

Bibliography

Akiake, Hirotugu (1973). “Information theory and an extension of the maximum likelihood principle”. In: *2nd International Symposium on Information Theory* (page 54).

Amari, Shun-ichi (1985). *Differential-Geometrical Methods in Statistics*. Lecture Notes in Statistics 28. Springer (page 6).

Andoni, Alexandr and Ilya Razenshteyn (2015). “Optimal Data-Dependent Hashing for Approximate Near Neighbors”. In: *ACM Symposium on Theory of Computing*. arXiv: 1501.01062 (page 12).

Ansolabehere, Stephen and Jonathan Rodden (2011). *Pennsylvania Data Files*. URL: http://hdl.handle.net/1902.1/16389 (page 84).

Auer, Peter, Nicoló Cesa-Bianchi, and Paul Fischer (2002). “Finite-time Analysis of the Multiarmed Bandit Problem”. In: *Machine Learning* 47, pages 235–256 (page 76).

Bach, Francis (2015). “On the Equivalence between Quadrature Rules and Random Features”. In: arXiv: arXiv:1502.06800 (pages 29, 31, 32).

Bardenet, Rémi and Odalric-Ambrym Maillard (2015). “Concentration inequalities for sampling without replacement”. In: *Bernoulli* 21.3, pages 1361–1385. arXiv: 1309.4029 (page 45).

Belongie, Serge, Charless Fowlkes, Fan Chung, and Jitendra Malik (2002). “Spectral partitioning with indefinite kernels using the Nyström extension”. In: *ECCV*. Springer, pages 531–542 (page 17).

Bengio, Yoshua, Jean-François Paiement, Pascal Vincent, Olivier Delalleau, Nicolas Le Roux, and Marie Ouimet (2004). “Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering”. In: *Advances in Neural Information Processing Systems*. NIPS (page 16).

Berlinet, Alain and Christine Thomas-Agnan (2004). *Reproducing kernel Hilbert spaces in Probability and Statistics*. Kluwer Academic Publishers (page 9).

Bernstein, Sergei (1924). “On a modification of Chebyshev’s inequality and of the error formula of Laplace”. Russian. In: *Ann. Sci. Inst. Savantes Ukraine, Sect. Math.* 1, pages 38–49 (page 25).

Betancourt, Michael (2015). “Adiabatic Monte Carlo”. Version 5. In: arXiv: 1405.3489v5 (page 47).

Beygelzimer, Alina, Sham Kakade, and John Langford (2006). “Cover trees for nearest neighbor”. In: *International Conference on Machine Learning*, pages 97–104 (page 12).

Bochner, Salomon (1959). *Lectures on Fourier integrals*. Princeton University Press (page 19).

Boiman, Oren, Eli Shechtman, and Michal Irani (2008). “In defense of nearest-neighbor based image classification”. In: *Computer Vision and Pattern Recognition* (page 5).

Bosch, Anna, Andrew Zisserman, and Xavier Muñoz (2006). “Scene classification via pLSA”. In: *ECCV* (page 56).

— (2008). “Scene Classification Using a Hybrid Generative/Discriminative Approach”. In: *IEEE Trans. PAMI* 30.4 (page 56).

Boucheron, Stéphane, Gábor Lugosi, and Pascal Massart (2013). *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford, UK: Oxford University Press (pages 27, 28, 104, 105).

Bounliphone, Wacha, Eugene Belilovsky, Matthew B. Blaschko, Ioannis Antonoglou, and Arthur Gretton (2015). “A Test of Relative Similarity For Model Selection in Generative Models”. In: arXiv: 1511.04581 (page 64).

Bousquet, Olivier (2002). “A Bennett concentration inequality and its application to suprema of empirical processes”. In: *Comptes Rendus Mathematique* 334, pages 495–500 (page 27).

Bousquet, Olivier and André Elisseeff (2001). “Algorithmic Stability and Generalization Performance”. In: *Advances in Neural Information Processing Systems*, pages 196–202 (page 31).

Bretagnolle, Jean, Didier Dacunha Castelle, and Jean-Louis Krivine (1966). “Lois stables et espaces L^P”. French. In: *Annales de l’institut Henri Poincaré (B) Probabilités et Statistiques* 2 (3), pages 231–259 (page 14).

Brochu, Eric, Vlad M Cora, and Nando de Freitas (2010). *A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning*. arXiv: 1012.2599 (page 76).

Brown, Jason L, Alison Cameron, Anne D Yoder, and Miguel Vences (2014). “A necessarily complex model to explain the biogeography of the amphibians and reptiles of Madagascar.” In: *Nature communications* 5, page 5046 (page 75).

Bubeck, Sébastien, Rémi Munos, and Gilles Stoltz (2010). “Pure exploration in multi-armed bandits problems”. In: *Algorithmic Learning Theory*, pages 23–37. arXiv: 0802.2655 (page 93).

Cha, Sung Hyuk and Sargur N. Srihari (2002). “On measuring the distance between histograms”. In: *Pattern Recognition* 35.6, pages 1355–1370 (page 7).

Chen, Yihua, Eric K Garcia, Maya R Gupta, Ali Rahimi, and Luca Cazzanti (2009). “Similarity-based classification: Concepts and algorithms”. In: *Journal of Machine Learning Research* 10, pages 747–776 (pages 15, 17).

Cheng, Steve (2013). *Differentiation under the integral sign*. Version 16. URL: http://planetmath.org/differentiationundertheintegralsign (page 99).

Chwialkowski, Kacper, Aaditya Ramdas, Dino Sejdinovic, and Arthur Gretton (2015). “Fast Two-Sample Testing with Analytic Representations of Probability Measures”. In: arXiv: 1506.04725 (pages 37, 63, 67).

Collobert, Ronan and Jason Weston (2008). “A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning”. In: *ICML* (page 92).

Cortes, Corinna, M Mohri, and A Talwalkar (2010). “On the impact of kernel approximation on learning accuracy”. In: *International Conference on Artificial Intelligence and Statistics*, pages 113–120 (pages 30–32).

Cressie, Noel and Timothy R.C. Read (1984). “Multinomial Goodness-of-fit Tests”. In: *Journal of the Royal Statistical Society, Series B* 46.3, pages 440–464 (page 9).

Csiszár, I. (1963). “Eine informationstheoretische Ungleichung und ihre Anwendung auf den Beweis der Ergodizitat von Markoffschen Ketten”. German. In: *Magyar. Tud. Akad. Mat. Kutato Int. Kozl* 8, pages 85–108 (page 6).

Cucker, Felipe and Steve Smale (2001). “On the mathematical foundations of learning”. In: *Bulletin of the American Mathematical Society* 39.1, pages 1–49 (pages 99, 105).

Cuturi, Marco (2013). “Sinkhorn distances: Lightspeed computation of optimal transport”. In: *Advances in Neural Information Processing Systems*. arXiv: arXiv:1306.0895v1 (pages 9, 12).

Daróczy, Zoltán (1970). “Generalized information functions”. In: *Information and Control* 16.1, pages 36–51 (page 8).

DLMF: NIST Digital Library of Mathematical Functions. Online companion to Olver, Lozier, Boisvert, and Clark 2010. URL: http://dlmf.nist.gov/ (pages 22, 119).

Dudley, Richard M (1967). “The sizes of compact subsets of Hilbert space and continuity of Gaussian processes”. In: *Journal of Functional Analysis* 1.3, pages 290–330 (pages 26, 27).

Dziugaite, Gintare Karolina, Daniel M. Roy, and Zoubin Ghahramani (2015). “Training generative neural networks via Maximum Mean Discrepancy optimization”. In: *Uncertainty in Artificial Intelligence*. arXiv: 1505.03906 (pages 61, 62, 90).

Edwards, D. A. (2011). “On the Kantorovich-Rubinstein theorem”. In: *Expositiones Mathematicae* 29.4, pages 387–398 (page 9).

Eisenstein, Daniel J., David H. Weinberg, Eric Agol, et al. (2011). “SDSS-III: Massive Spectroscopic Surveys of the Distant Universe, the Milky Way, and Extra-Solar Planetary Systems”. In: *The Astronomical Journal* 142, 72, page 72. arXiv: 1101.1529 (page 75).

El Alaoui, Ahmed and Michael W. Mahoney (2015). “Fast Randomized Kernel Ridge Regression With Statistical Guarantees”. In: *Advances in Neural Information Processing Systems*. arXiv: 1411.0306 (pages 16, 89).

Flaxman, Seth R., Dino Sejdinovic, John P. Cunningham, and Sarah Filippi (2016). “Bayesian Learning of Kernel Embeddings”. In: *Uncertainty in Artificial Intelligence*. arXiv: 1603.02160 (page 66).

Flaxman, Seth R., Yu-Xiang Wang, and Alexander J. Smola (2015). “Who Supported Obama in 2012? Ecological Inference through Distribution Regression”. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’15*. ACM Press, pages 289–298 (pages 1, 37).

Fuglede, Bent (2005). “Spirals in Hilbert space: With an application in information theory”. In: *Expositiones Mathematicae* 23.1, pages 23–45 (pages 14, 45, 46, 49).

Fukumizu, Kenji, Arthur Gretton, Xiaohai Sun, and Bernhard Schölkopf (2008). “Kernel Measures of Conditional Dependence”. In: *Advances in Neural Information Processing Systems*. Volume 20 (page 61).

Gardner, Andrew, Christian a. Duncan, Jinko Kanno, and Rastko R. Selmic (2015). “Earth Mover’s Distance Yields Positive Definite Kernels For Certain Ground Distances”. In: arXiv: 1510.02833 (page 14).

Garnett, Roman, Yamuna Krishnamurthy, Xuehan Xiong, Jeff Schneider, and Richard P Mann (2012). “Bayesian Optimal Active Search and Surveying”. In: *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)* (pages 76, 77).

Giné, Evarist and Armelle Guillou (2002). “Rates of strong uniform consistency for multivariate kernel density estimators”. In: *Ann. Inst. H. Poincaré Probab. Statist.* 38.6, pages 907–921 (pages 48, 109, 110).

Gönen, Mehmet and Ethem Alpaydın (2011). “Multiple Kernel Learning Algorithms”. In: *Journal of Machine Learning Research* 12, pages 2211–2268 (page 61).

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, et al. (2014). “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems 27*, pages 2672–2680. arXiv: arXiv:1406.2661v1 (page 90).

Gotovos, Alkis, Nathalie Casati, Gregory Hitz, and Andreas Krause (2013). “Active Learning for Level Set Estimation”. In: *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)* (pages 76, 82).

Grauman, Kristen and Trevor Darrell (2007). “The Pyramid Match Kernel: Efficient Learning with Sets of Features”. In: *JMLR* 8, pages 725–760 (pages 12, 56).

Gretton, Arthur, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alex J Smola (2012). “A Kernel Two-Sample Test”. In: *The Journal of Machine Learning Research* 13 (pages 9, 10, 13, 39, 40, 61, 62, 64, 68).

Gretton, Arthur, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf (2005). “Measuring statistical dependence with Hilbert-Schmidt norms”. In: *Algorithmic Learning Theory* (page 61).

Gretton, Arthur, Kenji Fukumizu, Zaid Harchaoui, and Bharath K. Sriperumbudur (2009). "A fast, consistent kernel two-sample test". In: *Advances in Neural Information Processing Systems*. MIT Press (page 68).

Gretton, Arthur and László Györfi (2010). "Consistent Nonparametric Tests of Independence". In: *Journal of Machine Learning Research* 11.172, pages 1391–1423 (page 12).

Gretton, Arthur, Bharath K. Sriperumbudur, Dino Sejdinovic, Heiko Strathmann, and Massimiliano Pontil (2012). "Optimal kernel choice for large-scale two-sample tests". In: *Advances in Neural Information Processing Systems*. Volume 25, pages 1214–1222 (pages 66–68, 72).

Grisel, Olivier, Andreas Mueller, Fabian Pedregosa, et al. (2016). *scikit-learn: 0.17.1 release tag for DOI*. DOI: 10.5281/zenodo.49911 (pages 9, 20, 95).

Haasdonk, Bernard and Claus Bahlmann (2004). "Learning with Distance Substitution Kernels". In: *Pattern Recognition: 26th DAGM Symposium*, pages 220–227 (page 13).

Harris, Z. (1954). "Distributional structure". In: *Word* 10.23, pages 146–162 (page 93).

Havrda, Jan and František Charvát (1967). "Quantification method of classification processes". Czech. In: *Kybernetika (Prague)* 3, pages 30–35 (page 8).

Hino, Hideitsu and Noboru Murata (2013). "Information estimators for weighted observations". In: *Neural Networks* 46, pages 260–275 (page 96).

Hoeffding, Wassily (1963). "Probability inequalities for sums of bounded random variables". In: *Journal of the American Statistical Association* 58.301, pages 13–30 (page 25).

Ismail, Mourad E. H. (1990). "Complete monotonicity of modified Bessel functions". In: *Proceedings of the American Mathematical Society* 108.2, pages 353–361 (page 22).

Jebara, T., R. Kondor, A. Howard, K. Bennett, and N. Cesa-bianchi (2004). "Probability product kernels". In: *JMLR* 5, pages 819–844 (page 10).

Jin, Jay (2016). "Detection of Sources of Harmful Radiation using Portable Sensors". M.Sc. Carnegie Mellon University. Technical Report CMU-CS-16-115 (pages 3, 59).

Jin, Jay, Kyle Miller, Dougal J. Sutherland, Simon Labov, Karl Nelson, and Artur Dubrawski (2016). "List Mode Regression for Low Count Detection". To be presented as a poster at the 2016 IEEE Nuclear Science Symposium. URL: https://event.crowdcompass.com/2016-nss-mic/activity/8Jh9tYvmAt (pages 3, 51).

Jitkrittum, Wittawat, Arthur Gretton, Nicolas Heess, S M Ali Eslami, Balaji Lakshminarayanan, Dino Sejdinovic, and Zoltán Szabó (2015). "Kernel-Based Just-In-Time Learning for Passing Expectation Propagation Messages". In: *Uncertainty in Artificial Intelligence* (pages 1, 37, 54).

Jitkrittum, Wittawat, Zoltán Szabó, Kacper Chwialkowski, and Arthur Gretton (2016). "Interpretable Distribution Features with Maximum Testing Power". In: *International Conference on Machine Learning*. arXiv: 1605.06796 (page 67).

Karayev, Sergey, Matthew Trentacoste, Helen Han, Aseem Agarwala, Trevor Darrell, Aaron Hertzmann, and Holger Winnemoeller (2013). "Recognizing Image Style". In: *arXiv:1311.3715*. arXiv: 1311.3715 (page 91).

Khosravifard, Mohammadali, Dariush Fooladivanda, and T. Aaron Gulliver (2007). "Confliction of the Convexity and Metric Properties in f-Divergences". In: *IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Sciences* E90-A.9, pages 1848–1853 (page 6).

Klypin, Anatoly, Gustavo Yepes, Stefan Gottlober, Francisco Prada, and Steffen Hess (2014). "MultiDark simulations: the story of dark matter halo concentrations and density profiles". In: arXiv: 1411.4001 (page 52).

Krishnamurthy, Akshay, Kirthevasan Kandasamy, Barnabás Póczos, and Larry Wasserman (2014). "Nonparametric Estimation of Rényi Divergence and Friends". In: *International Conference on Machine Learning*. arXiv: 1402.2966 (pages 12, 110).

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances In Neural Information Processing Systems*. arXiv: 1102.0183 (pages 57, 91).

Kroemer, O. B., R. Detry, J. Piater, and J. Peters (2010). "Combining active learning and reactive control for robot grasping". In: *Robotics and Autonomous Systems* 58.9, pages 1105–1116 (page 76).

Kusner, Matt, Yu Sun, Nicholas Kolkin, and Kilian Weinberger (2015). "From Word Embeddings To Document Distances". In: *Proceedings of The 32nd International Conference on Machine Learning*, pages 957–966 (pages 5, 92).

Lafferty, John, Han Liu, and Larry Wasserman (2012). "Sparse Nonparametric Graphical Models". In: *Statistical Science* 27.4, pages 519–537. arXiv: 1201.0794 (page 47).

Lan, Shiwei, Jeffrey Streets, and Babak Shahbaba (2014). "Wormhole Hamiltonian Monte Carlo". In: *AAAI Conference on Artificial Intelligence*. arXiv: 1306.0063 (page 47).

Lazebnik, Svetlana, Cordelia Schmid, and Jean Ponce (2006). "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories". In: *CVPR* (page 57).

Le, Quoc, Tamás Sarlós, and Alex J Smola (2013). "Fastfood — Approximating Kernel Expansions in Loglinear Time". In: *International Conference on Machine Learning*. arXiv: 1408.3060 (page 92).

Leung, Thomas and Jitendra Malik (2001). "Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons". In: *IJCV* 43, pages 29–44 (page 12).

Levy, Omer and Yoav Goldberg (2014). "Neural Word Embedding as Implicit Matrix Factorization". In: *Advances in Neural Information Processing Systems*, pages 2177–2185 (page 93).

Li, Shukai and Ivor W Tsang (2011). "Learning to Locate Relative Outliers". In: *Asian Conference on Machine Learning*. Volume 20. JMLR: Workshop and Conference Proceedings, pages 47–62 (page 37).

Li, Yujia, Kevin Swersky, and Richard Zemel (2015). "Generative Moment Matching Networks". In: *Uncertainty in Artificial Intelligence*. arXiv: 1502.02761 (pages 61, 90).

Liese, Friedrich and Igor Vajda (2006). "On divergences and informations in statistics and information theory". In: *IEEE Transactions on Information Theory* 52.10, pages 4394–4412 (pages 6, 9).

Lin, Jianhua (1991). "Divergence measures based on the Shannon entropy". In: *IEEE Transactions on Information Theory* 37.1, pages 145–151 (page 110).

Lin, Min, Qiang Chen, and Shuicheng Yan (2014). "Network in network". In: *ICLR*. arXiv: 1312.4400 (page 91).

Lopez-Paz, David, Krikamol Muandet, Bernhard Schölkopf, and Ilya Tolstikhin (2015). "Towards a Learning Theory of Cause-Effect Inference". In: *ICML*. arXiv: 1502.02398 (pages 1, 37).

Low, Kian Hsiang, Jie Chen, John M. Dolan, Steve Chien, and David R. Thompson (2012). "Decentralized Active Robotic Exploration and Mapping for Probabilistic Field Classification in Environmental Sensing". In: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*. AAMAS '12. Valencia, Spain, pages 105–112 (page 76).

Lowe, David G. (2004). "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* 60.2, pages 91–110 (page 56).

Ma, Yifei, Roman Garnett, and Jeff Schneider (2014). "Active Area Search via Bayesian Quadrature". In: *Seventeenth International Conference on Artificial Intelligence and Statistics*. AISTATS (pages 2, 76, 79, 82).

Ma, Yifei, Dougal J. Sutherland, Roman Garnett, and Jeff Schneider (2015). “Active Pointillistic Pattern Search”. In: *Eighteenth International Conference on Artificial Intelligence and Statistics*. AISTATS (pages 3, 81).

Martins, André F. T., Noah A. Smith, Eric P. Xing, Pedro M. Q. Aguiar, and Mário A. T. Figueiredo (2009). “Nonextensive Information Theoretic Kernels on Measures”. In: *The Journal of Machine Learning Research* 10 (page 8).

McDiarmid, Colin (1989). “On the method of bounded differences”. In: *Surveys in combinatorics* 141.1, pages 148–188 (page 97).

Mehta, Nishant A. and Alexander G. Gray (2010). “Generative and Latent Mean Map Kernels”. In: arXiv: 1005.0188 (page 37).

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). “Distributed Representations of Words and Phrases and their Compositionality”. In: *Advances in Neural Information Processing Systems* (pages 92, 93).

Mitchell, Lee J., Bernard F. Phlips, W. Neil Johnson, et al. (2009). “Mobile Imaging and Spectroscopic Threat Identification (MISTI): System overview”. In: *IEEE Nuclear Science Symposium Conference Record*, pages 110–118 (page 59).

Moon, Kevin R. and Alfred O. Hero (2014a). “Ensemble estimation of multivariate f-divergence”. In: *2014 IEEE International Symposium on Information Theory*. IEEE, pages 356–360. arXiv: 1404.6230 (page 13).

— (2014b). “Multivariate f-divergence Estimation With Confidence”. In: *Advances in Neural Information Processing Systems*, pages 2420–2428 (page 13).

Moreno, Pedro J., Purdy P. Ho, and Nuno Vasconcelos (2004). “A Kullback-Leibler Divergence Based Kernel for SVM Classification in Multimedia Applications”. In: *NIPS* (page 10).

Muandet, Krikamol, Kenji Fukumizu, Bharath K. Sriperumbudur, Arthur Gretton, and Bernhard Schölkopf (2014). “Kernel Mean Estimation and Stein’s Effect”. In: *International Conference on Machine Learning*. arXiv: arXiv:1306.0842v2 (page 13).

Muandet, Krikamol, Bernhard Schölkopf, Kenji Fukumizu, and Francesco Dinuzzo (2012). “Learning from Distributions via Support Measure Machines”. In: *Advances in Neural Information Processing Systems*. arXiv: arXiv:1202.6504v2 (pages 10, 48).

Muja, Marius and David G. Lowe (2009). “Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration”. In: *International Conference on Computer Vision Theory and Applications (VISAPP ’09)* (page 12).

Müller, Alfred (1997). “Integral Probability Metrics and their Generating Classes of Functions”. In: *Advances in Applied Probability* 29.2, pages 429–443 (pages 6, 7).

Naidan, Bilegsaikhan, Leonid Boytsov, and Eric Nyberg (2015). “Permutation Search Methods are Efficient, Yet Faster Search is Possible”. In: *Proceedings of the 41st International Conference on Very Large Data Bases*, pages 1618–1629. arXiv: 1506.03163 (page 12).

Naor, Assaf and Gideon Schechtman (2007). “Planar Earthmover is not in L_1”. In: *SIAM Journal on Computing* 37.3, pages 804–826 (page 14).

Nguyen, Xuanlong, Martin J. Wainwright, and Michael I. Jordan (2010). “Estimating divergence functionals and the likelihood ratio by convex risk minimization”. In: *IEEE Transactions on Information Theory* 56.11, pages 5847–5861. arXiv: 0809.0853 (page 13).

Nielsen, Frank and Richard Nock (2011). “On Rényi and Tsallis entropies and divergences for exponential families”. In: arXiv: 1105.3259 (page 10).

Niranjan, Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger (2010). “Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design”. In: *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)* (page 76).

Ntampaka, Michelle, Hy Trac, Dougal J. Sutherland, Nicholas Battaglia, Barnabás Póczos, and Jeff Schneider (2015). “A Machine Learning Approach for Dynamical Mass Measurements of Galaxy Clusters”. In: *The Astrophysical Journal* 803.2, page 50. arXiv: 1410.0686 (pages 3, 51–53).

Ntampaka, Michelle, Hy Trac, Dougal J. Sutherland, Sebastian Fromenteau, Barnabás Póczos, and Jeff Schneider (in press). “Dynamical Mass Measurements of Contaminated Galaxy Clusters Using Machine Learning”. In: *The Astrophysical Journal*. arXiv: 1509.05409. In press (pages 3, 51, 53).

Oliva, Aude and Antonio Torralba (2001). “Modeling the shape of the scene: a holistic representation of the spatial envelope”. In: *International Journal of Computer Vision* 42.3 (page 56).

Oliva, Junier B., Avinava Dubey, Barnabas Poczos, Jeff Schneider, and Eric P Xing (2016). “Bayesian Nonparametric Kernel-Learning”. In: *AISTATS*. arXiv: 1506.08776 (page 61).

Oliva, Junier B., Willie Neiswanger, Barnabás Póczos, Jeff Schneider, and Eric Xing (2014). “Fast Distribution To Real Regression”. In: *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS)*. arXiv: 1311.2236 (pages 43, 44, 54).

Oliva, Junier B., Barnabás Póczos, and Jeff Schneider (2013). “Distribution to distribution regression”. In: *Proceedings of The 30th International Conference on Machine Learning* (page 5).

Oliva, Junier B., Dougal J. Sutherland, Barnabás Póczos, and Jeff Schneider (2015). “Deep Mean Maps”. In: arXiv: 1511.04150 (pages 91, 92).

Olver, F. W. J., D. W. Lozier, R. F. Boisvert, and C. W. Clark, editors (2010). *NIST Handbook of Mathematical Functions*. Print companion to DLMF. New York, NY: Cambridge University Press (page 114).

Osborne, Michael A., Roman Garnett, and Stephen J. Roberts (2009). “Gaussian Processes for Global Optimization”. In: *Proceedings of the 3rd Learning and Intelligent Optimization Conference (LION 3)* (page 76).

Perlman, Eric, Randal Burns, Yi Li, and Charles Meneveau (2007). “Data Exploration of Turbulence Simulations using a Database Cluster”. In: *Proceedings of the 2007 ACM/IEEE Conference on Supercomupting* (page 85).

Póczos, Barnabás, Alessandro Rinaldo, Aarti Singh, and Larry Wasserman (2013). “Distribution-Free Distribution Regression”. In: *Artificial Intelligence and Statistics*. AISTATS. arXiv: 1302.0082 (page 5).

Póczos, Barnabás and Jeff Schneider (2011). “On the Estimation of α-Divergences”. In: *International Conference on Artificial Intelligence and Statistics* (pages 13, 96).

Póczos, Barnabás, Liang Xiong, and Jeff Schneider (2011). “Nonparametric Divergence Estimation with Applications to Machine Learning on Distributions”. In: *Uncertainty in Artificial Intelligence* (page 5).

Póczos, Barnabás, Liang Xiong, Dougal J. Sutherland, and Jeff Schneider (2012). “Nonparametric kernel estimators for image classification”. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2989–2996 (pages 2, 6, 13, 51, 56, 96).

Puzicha, J., T. Hofmann, and J.M. Buhmann (1997). “Non-parametric similarity measures for unsupervised texture segmentation and image retrieval”. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* March 2016, pages 267–272 (page 9).

Qin, Jianzhao and Nelson H.C. Yung (2010). “SIFT and color feature fusion using localized maximum-margin learning for scene classfication”. In: *International Conference on Machine Vision* (page 56).

Quiter, Brian J., Lavanya Ramakrishnan, and Mark S. Bandstra (2015). *GRDC: A Collaborative Framework for Radiological Background and Contextual Data Analysis*. Technical report. Berkeley, CA (United

States): Lawrence Berkeley National Laboratory (LBNL). URL: http://www.osti.gov/servlets/purl/1235086/ (page 59).

Raff, Edward (2011-16). *JSAT: Java Statistical Analysis Tool*. https://github.com/EdwardRaff/JSAT/ (page 20).

Rahimi, Ali and Benjamin Recht (2007). "Random Features for Large-Scale Kernel Machines". In: *Advances in Neural Information Processing Systems*. MIT Press (pages 2, 19, 20, 24–26, 34, 99, 101).

— (2008a). "Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning". In: *Advances in Neural Information Processing Systems*. MIT Press, pages 1313–1320 (pages 20, 29, 31, 32).

— (2008b). "Uniform approximation of functions with random bases". In: *46th Annual Allerton Conference on Communication, Control, and Computing*, pages 555–561 (pages 20, 29).

Ramdas, Aaditya, Sashank J. Reddi, Barnabas Poczos, Aarti Singh, and Larry Wasserman (2015). "Adaptivity and Computation-Statistics Tradeoffs for Kernel and Distance based High Dimensional Two Sample Testing". In: arXiv: 1508.00655 (pages 62, 63, 65).

Ramdas, Aaditya and Leila Wehbe (2015). "Nonparametric Independence Testing for Small Sample Sizes". In: arXiv: 1406.1922 (page 13).

Rasmussen, Carl Edward and Zoubin Ghahramani (2003). "Bayesian Monte Carlo". In: *Advances in Neural Information Processing Systems 15 (NIPS 2002)* (page 79).

Reddi, Sashank J., Aaditya Ramdas, Barnabás Póczos, Aarti Singh, and Larry Wasserman (2014). "On the Decreasing Power of Kernel and Distance based Nonparametric Hypothesis Tests in High Dimensions". In: *AAAI Conference on Artificial Intelligence*. arXiv: 1406.2083 (page 65).

Rényi, Alfréd (1961). "On measures of entropy and information". In: *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 547–561 (page 8).

Rubner, Yossi, Carlo Tomasi, and Leonidas J. Guibas (2000). "Earth mover's distance as a metric for image retrieval". In: *International Journal of Computer Vision* 40.2, pages 99–121 (pages 9, 12).

Rudi, Alessandro, Raffaello Camoriano, and Lorenzo Rosasco (2015). "Less is More: Nyström Computational Regularization". In: *Advances in Neural Information Processing Systems*. arXiv: 1507.04717 (page 17).

— (2016). "Generalization Properties of Learning with Random Features". In: arXiv: 1602.04474 (pages 29, 31, 89).

Russakovsky, Olga, Jia Deng, Hao Su, et al. (2014). "Imagenet large scale visual recognition challenge". In: *International Journal of Computer Vision*, pages 1–42 (page 91).

Saunders, C., A. Gammerman, and V. Vovk (1998). "Ridge Regression Learning Algorithm in Dual Variables". In: *Proceedings of the 15th International Conference on Machine Learning*, pages 515–521 (page 30).

Schoenberg, I. J. (1938). "Metric spaces and positive definite functions". In: *Transactions of the American Mathematical Society* 44.3, pages 522–536 (page 13).

Schwarz, Gideon (1978). "Estimating the Dimension of a Model". In: *Ann. Statist.* 6.2, pages 461–464 (page 54).

Serfling, Robert J. (1974). "Probability Inequalities for the Sum in Sampling without Replacement". In: *The Annals of Statistics* 2.1, pages 39–48 (page 45).

— (1980). *Approximation Theorems of Mathematical Statistics*. John Wiley & Sons (pages 62, 64).

Settles, Burr (2012). *Active Learning*. Morgan & Claypool (page 75).

Shirdhonkar, Sameer and David W. Jacobs (2008). "Approximate earth mover's distance in linear time". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8 (page 12).

Simonyan, Karen and Andrew Zisserman (2015). “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *ICLR*. arXiv: 1409.1556 (page 57).

Singh, Shashank and Barnabás Póczos (2014). “Exponential Concentration of a Density Functional Estimator”. In: *Advances in Neural Information Processing Systems*, pages 3032–3040 (page 12).

SKBMoore (2016). *Decay relationship with modified Bessel functions of the second kind*. Version 2016-08-29. MathOverflow: a/248510 (page 22).

Song, Le, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt (2012). “Feature Selection via Dependence Maximization”. In: *Journal of Machine Learning Research* 13, pages 1393–1434 (page 61).

Sonnenburg, Sören, Gunnar Raetsch, Sebastian Henschel, et al. (2010). “The SHOGUN Machine Learning Toolbox”. In: *Journal of Machine Learning Research* 11, pages 1799–1802 (pages 20, 68, 69).

Sriperumbudur, Bharath K., Kenji Fukumizu, Arthur Gretton, Gert R. G. Lanckriet, and Bernhard Schölkopf (2009). “Kernel choice and classifiability for RKHS embeddings of probability distributions”. In: *Advances in Neural Information Processing Systems*. Volume 22. MIT Press, pages 1750–1758 (pages 61, 66).

Sriperumbudur, Bharath K., Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert R. G. Lanckriet (2009). “On integral probability metrics, ϕ-divergences and binary classification”. In: arXiv: 0901.2698 (page 6).

— (2012). “On the empirical estimation of integral probability metrics”. In: *Electronic Journal of Statistics* 6, pages 1550–1599 (page 13).

Sriperumbudur, Bharath K., Kenji Fukumizu, Revant Kumar, Arthur Gretton, and Aapo Hyvärinen (2013). “Density Estimation in Infinite Dimensional Exponential Families”. In: arXiv: 1312.3516 (page 47).

Sriperumbudur, Bharath K., Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert R. G. Lanckriet (2010). “Hilbert space embeddings and metrics on probability measures”. In: *Journal of Machine Learning Research* 11, pages 1517–1561. arXiv: 0907.5309 (pages 9, 10, 45, 49).

Sriperumbudur, Bharath K. and Zoltán Szabó (2015). “Optimal Rates for Random Fourier Features”. In: arXiv: 1506.02155 (pages 24, 29).

Stein, Charles (1956). “Inadmissibility of the Usual Estimator for the Mean of a Multi-Variate Normal Distribution”. In: *Proc. Third Berkeley Symp. Math. Statist. Prob* 1.4, pages 197–206 (page 13).

Strathmann, Heiko (2012). “Adaptive Large-Scale Kernel Two-Sample Testing”. M.Sc. University College London. URL: http://herrstrathmann.de/wp-content/uploads/2012/09/2012_Strathmann_MSc.pdf (pages 66, 67).

Sugiyama, Masashi, Taiji Suzuki, Yuta Itoh, Takafumi Kanamori, and Manabu Kimura (2011). “Least-squares two-sample test”. In: *Neural Networks* 24.7, pages 735–751 (page 66).

Sutherland, Dougal J. (2015). *Earth Mover’s Distance (EMD) between two Gaussians*. Version 2015-04-23. CrossValidated: a/144896 (page 10).

Sutherland, Dougal J., Junier B. Oliva, Barnabás Póczos, and Jeff Schneider (2016). “Linear-Time Learning on Distributions with Approximate Kernel Embeddings”. In: *AAAI Conference on Artificial Intelligence*. arXiv: 1509.07553 (pages 2, 3, 37, 45, 51).

Sutherland, Dougal J. and Jeff Schneider (2015). “On the Error of Random Fourier Features”. In: *Uncertainty in Artificial Intelligence*. arXiv: 1506.02785 (pages 2, 20, 37).

Sutherland, Dougal J., Liang Xiong, Barnabás Póczos, and Jeff Schneider (2012). “Kernels on Sample Sets via Nonparametric Divergence Estimates”. In: arXiv: 1202.0302 (pages 2, 51, 85, 86).

Szabó, Zoltán, Bharath K. Sriperumbudur, Barnabás Póczos, and Arthur Gretton (2015). “Learning Theory for Distribution Regression”. In: *Artificial Intelligence and Statistics*. AISTATS. arXiv: 1411.2066 (pages 10, 89).

Szegedy, Christian, Wei Liu, Yangqing Jia, et al. (2014). “Going Deeper with Convolutions”. In: arXiv: 1409.4842 (page 91).

Tao, Chenyang and Jianfeng Feng (2016). “Nonlinear association criterion, nonlinear Granger causality and related issues with applications to neuroimage studies”. In: *Journal of Neuroscience Methods* 262, pages 110–132 (page 61).

Tesch, Matthew, Jeff Schneider, and Howie Choset (2013). “Expensive function optimization with stochastic binary outcomes”. In: *Proceedings of the 30th International Conference on Machine Learning (ICML 2013)* (page 76).

The Theano Development Team, Rami Al-Rfou, Guillaume Alain, et al. (2016). *Theano: A Python framework for fast computation of mathematical expressions*. arXiv: 1605.02688 (page 69).

Topsøe, Flemming (2000). “Some inequalities for information divergence and related measures of discrimination”. In: *IEEE Transactions on Information Theory* 46.4, pages 1602–1609 (page 14).

Tsallis, Constantino (1988). “Possible generalization of Boltzmann-Gibbs statistics”. In: *Journal of Statistical Physics* 52.1-2, pages 479–487 (page 8).

Turian, Joseph, Lev Ratinov, and Yoshua Bengio (2010). “Word representations: A simple and general method for semi-supervised learning”. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (page 92).

United States Census Bureau (2010). *2010 Census*. URL: http://www.census.gov/2010census/data/ (page 84).

Valada, Abhinav, Christopher Tomaszewski, Balajee Kannan, Prasanna Velagapudi, George Kantor, and Paul Scerri (2012). “An Intelligent Approach to Hysteresis Compensation while Sampling Using a Fleet of Autonomous Watercraft”. In: *Intelligent Robotics and Applications*. Volume 7507. Lecture Notes in Computer Science (pages 75, 81, 82).

Vedaldi, Andrea and Brian Fulkerson (2008). *VLFeat: An Open and Portable Library of Computer Vision Algorithms*. http://www.vlfeat.org/ (page 56).

Vedaldi, Andrea and Andrew Zisserman (2012). “Efficient additive kernels via explicit feature maps”. In: *IEEE transactions on pattern analysis and machine intelligence* 34.3, pages 480–92 (pages 9, 14, 46, 54).

Vilnis, Luke and Andrew McCallum (2015). “Word Representations via Gaussian Embedding”. In: *International Conference on Learning Representations*. arXiv: 1412.6623 (page 93).

Wang, Fei, Tanveer Syeda-Mahmood, Baba C. Vemuri, David Beymer, and Anand Rangarajan (2009). “Closed-Form Jensen-Renyi Divergence for Mixture of Gaussians and Applications to Group-Wise Shape Registration”. In: *Med Image Comput Comput Assist Interv.* 12.1, pages 648–655 (page 10).

Wang, Qing, Sanjeev R Kulkarni, and Sergio Verdú (2009). “Divergence Estimation for Multidimensional Densities Via k-Nearest-Neighbor Distances”. In: *IEEE Transactions on Information Theory* 55.5, pages 2392–2405 (pages 13, 52, 54, 96).

Wasserman, Larry (2006). *All of Nonparametric Statistics*, page 268 (page 12).

Williams, Christopher K I and Matthias Seeger (2000). “Using the Nyström method to speed up kernel machines”. In: *Advances in Neural Information Processing Systems*. MIT Press, pages 682–688 (page 16).

Wilson, Andrew Gordon, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing (2016). “Deep Kernel Learning”. In: *AISTATS*. arXiv: 1511.02222 (page 61).

Wu, Jianxin, Bin-Bin Gao, and Guoqing Liu (2016). “Visual Recognition Using Directional Distribution Distance”. In: *AAAI Conference on Artificial Intelligence*. arXiv: 1504.04792 (pages 57, 58).

Xiong, Liang (2013). “On Learning from Collective Data”. PhD thesis. Carnegie Mellon University (page 16).

Yang, Kun, Hao Su, and Wing Hung Wong (2014). “co-BPM: a Bayesian Model for Estimating Divergence and Distance of Distributions”. In: arXiv: 1410.0726 (page 13).

Yang, Tianbao, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou (2012). “Nyström Method vs Random Fourier Features: A Theoretical and Empirical Comparison”. In: *Advances in Neural Information Processing Systems*. MIT Press (pages 32, 89).

Yang, Zichao, Alexander J Smola, and Andrew Gordon Wilson (2015). “A la Carte — Learning Fast Kernels”. In: *AISTATS*. arXiv: 1412.6493 (pages 61, 92).

Yoshikawa, Yuya, Tomoharu Iwata, and Hiroshi Sawada (2014). “Latent Support Measure Machines for Bag-of-Words Data Classification”. In: *Advances in Neural Information Processing Systems*, pages 1961–1969 (pages 90, 92, 93).

— (2015). “Non-Linear Regression for Bag-of-Words Data via Gaussian Process Latent Variable Set Model”. In: *AAAI Conference on Artificial Intelligence*, pages 3129–3135 (pages 90, 92, 93).

Zhang, J, M Marszałek, S Lazebnik, and C Schmid (2006). “Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study”. In: *International Journal of Computer Vision* 73.2, pages 213–238 (pages 9, 14).

Zhao, Ji and Deyu Meng (2014). “FastMMD: Ensemble of Circular Discrepancy for Efficient Two-Sample Test”. In: arXiv: 1405.2664 (pages 37, 38).

Zhou, Bolei, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva (2014). “Learning Deep Features for Scene Recognition using Places Database”. In: *NIPS* (pages 58, 91).

Zwicky, Fritz (1933). “Die Rotverschiebung von extragalaktischen Nebeln”. German. In: *Helvetica Physica Acta* 6, pages 110–127 (page 51).